Human Action Recognition from Digital Videos Based on Deep Learning Methods

Chenwei Liang

A thesis submitted to the Auckland University of Technology
in partial fulfillment of the requirements for the degree of
Master of Computer and Information Sciences (MCIS)

2022

School of Engineering, Computer & Mathematical Sciences

Abstract

Surveillance today is widely used in public safety and security. Additionally, more and more people have their own smartphones, the advancement has led to a large amount of video data being generated every day. Recognizing human actions from digital videos can not only contribute to the public safety, but also transform a large amount of video data into usable information. Therefore, in this thesis, we propose a YOLOv7-based model that utilizes various attention mechanisms for human action recognition. In the options of attention mechanisms, we choose CBAM and SimAM attention as our main framework.

Based on these two attention mechanisms, in this thesis, we propose three models: YOLOv7+CBAM, YOLOv7+SimAM, and YOLOv7+CBAM+SimAM. The three models are able to recognize five human actions (i.e., clapping, punching, walking, waving, running). In addition, the dataset in this thesis is to select suitable data samples from six public datasets, we acquire these data samples that can be employed for YOLOv7 training and testing.

Finally, through this dataset, YOLOv7 using the attention mechanism improves the accuracy by 7% over the base model. After the experiment, the accuracy of YOLOv7+CBAM+SimAM model is the highest one, which is up to 99.6%. The computing speed has also been improved, which takes 295ms to process one video frame on average.

Keywords: Human action recognition, YOLOv7, SBAM attention mechanism, SimAM attention mechanism

Table of Contents

List of Figures

List of Tables

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgments), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signature: 梁宸玮 Date: November 2022

Acknowledgment

I would firstly like to express my sincerest gratitude to my supervisor Wei Qi Yan. I wouldn't be where I am today without his help during my masters study at Auckland University of Technology. Thanks to him for the professional support and technical guidance he provided me during my studies.

In addition, I would like to thank the Auckland University of Technology for its support during my studies during COVID-19, which allowed me to continue my studies smoothly. Finally, thanks to my parents for the financial support they offered me.

Chenwei LIANG

Auckland, New Zealand

November 2022

Chapter 1
Introduction

The first chapter of this thesis consists of five distinct parts. Among them, the first section introduces the background and motivation of this thesis through a brief introduction, in which we explain the applications of human action recognition in intelligent surveillance systems. In the second section of this chapter, we present the specific details of the research question in detail. In the third section, we describe the relevant contributions of this thesis. Building on the previous three sections, the research objectives of this thesis are stated in Section IV, and the structure of thesis is summarized in Section V.

1.1 Background and Motivation

Affected by the rapid development of monitoring equipment technology, many digital monitoring devices have been installed at every corner of our society (Kieran & Yan, 2010). The popularity and installation of these surveillance equipment have brought a large number of surveillance videos (Yan, 2019). There is a large amount of information in these surveillance videos. In order to facilitate extraction of latent values in the video data, relevant technologies in the field of computer vision were widely explored (Yan, Kieran, Rafatirad, & Jain, 2011).

Computer vision is abbreviated as CV which refers to a simulation of human vision that allows computers to use digital cameras or other related equipment (Xu, et al., 2021). This simulation not only means that what the computer sees is as same as what a person sees, it also means that the computer can understand what it sees in a picture or video. This means that the computer visually behaves like human beings. Therefore, the field of computer vision also belongs to a more specific application of artificial intelligence (Lemley, Bazrafkan, & Corcoran, 2017). Computer vision tasks encapsulate object classification (Bansal, Yan, & Kankanhalli, 2003), visual object detection (Shen, Chen, Nguyen, & Yan, Flame detection using deep learning, 2018), and image segmentation (Mail & Lucas, 2018). The realization of these tasks often relies on machine learning and further deep learning (Liu, Yan, & Yang, 2018). The use of machine learning, deep learning and other technologies in CV brings greater possibilities for the processing of complex visual signals, and further realizes more accurate target recognition, target tracking and other applications.

Various video surveillance systems are now widely existing at every corner of human society (Wang, Kankanhalli, Yan, & Jain, 2003), such as parking lots, supermarkets, banks, factories, mines, and other places often have a large number of monitoring equipment (Fan, et al., 2015). This owns to the continuous development of human society, the living standards of most human beings have been significantly enhanced compared with the past, which makes people pay much more attention on their own security issues.

This has led to the explosive proliferation of the need for video surveillance systems, but the conventional surveillance systems often fail to actively monitor and monitor video content in real time (Grasso & Schembra, 2018). They usually hire many security staff members to observe the surveillance video synchronously or to analyze the video information recorded in the surveillance system manually after abnormal behaviors and events occur. These videos can be used in flame detection (Shen, Chen, Nguyen, & Yan, 2018), license plate recognition (Wang, Bacic, & Yan, 2018), object tracking and detecting pedestrian trajectories (Wang, Zhang & Yan, 2020), identifying, and comparing visual objects (Zheng, Yan & Nand, 2017), recognizing human behavior (Wang & Yan, 2019), etc. The focus of this thesis is on the problems of human action recognition from video, which becomes increasingly important as the number of surveillance videos grows.

Traditional usages of surveillance videos to identify human action, most of the effective information can only be obtained by relying on considerable number of manual operations. This also brings up a great deal of problems. The energy of person is often very limited. As the working hours enhance, the energy and concentration of people will plummet. When a person's energy drops, he may not be able to notice the occurrence of aberrant situations at the first time, resulting in some undue losses. The methods that can automatically recognize human actions are now urgently needed as a result of this.

Computers automatically detect and recognize human actions through video footage or image sequences are generally referred to as human action recognition (Lu, Shen, Yan, & Bačić, 2018). The method of human action recognition enables continuous real-time recognition that runs continuously for 24 hours, as well as automatic and efficient analysis of the video data gathered. The applications for human action recognition are numerous. In terms of video surveillance, it is utilized in a variety of public settings, including prisons, courts, education, transportation, public security, and many others, in addition to banks, post offices, and telecommunications. Additionally, it is crucial in big warehouses and military locations (Tripathi, Jalal, & Agrawal, 2017).

Patient monitoring system (Gul, Yousaf, Nawaz, Rehman, & Kim, 2020), medical care (Wentao Hu1, Huang, Zhan, & Yang, 2020), human-computer interaction (Malibari, et al., 2022), virtual reality (Ma, 2021), smart home (Zamil, Rawashdeh, Karime, & Hossain, 2019), smart security (Rathod, et al., 2020), athlete-assisted training (Guo, Mu, Xiong, Liu, & Gu, 2019), motion capture (MathieuBarnachon, SaïdaBouakaz, BoubakeurBoufama, & ErwanGuilloua, 2014), environmental control and monitoring (Li-ming, Huang, & Tan, 2013), sports and entertainment analysis (Wu, et al., 2022) have all made extensive use of human action recognition.

In addition, human behavior recognition methods were also employed for content-based video indexing (Saoudi & Jai-Andaloussi, 2021), etc. To sum up, the visual analysis of human motion has great practical significance. On the other hand, due to the complexity of human movement and the variability of the external environment. The influence of environmental factors such as cluttered background, occlusion, and perspective changes. This makes human action recognition and detection still a very tough problem. It is also a very attractive and challenging problem (Khan, et al., 2020).

Relevant research work on the identification and analysis of human behavior dates to 1973. Johansson presented a human model in their experimental study (Johansson, 1973). In the model specifically, the behavior of the human body is split into 12 relevant points. This point model method for describing behavior plays a crucial guiding role in later action recognition algorithms based on human structure. Later, a series of techniques that can recognize human actions have been put forth. Although various methods can produce accurate recognition, the effectiveness of recognition has always been a challenge. This challenge was not resolved until the development of deep learning and related technologies, which utilizes deep learning and related technologies to human behavior recognition (Yan, 2021). This is a very effective concept, as demonstrated by subsequent technical advancement and reality. The use of deep learning methods for human action recognition greatly saves manpower. The emergence of these methods has greatly improved the efficiency of the monitoring system.

Deep learning is broadly employed for human action recognition, more and more action recognition is prone to maturity. For example, human action recognition using convolutional neural networks (Karpathy, et al., 2014) (Chauhan, Ghanshala, & Joshi, 2018) and recurrent neural networks (Ng, et al., 2015). Some issues also exist. The majority of deep learning methods for human action recognition now in use rely on additional feature extraction methods. People also need to select the right feature extraction strategies in addition to the necessity for deep learning techniques. People frequently must spend a lot of time through this. The time needed for recognition also enhances because the majority of deep learning approaches are unable to achieve end-to-end recognition. Therefore, a deep learning method that can be applied to fastly and swiftly action recognition becomes crucial.

The literature on this topic is constantly being produced in light of the impact of these aspects on the recognition of human behavior and the advancement of deep learning technology (An & Yan, 2021). This makes it possible for this thesis to conduct further in-depth research work based on past knowledge, thus, the research questions of this thesis are proposed.

1.2 Research Questions

In this thesis, we implemented human action recognition through surveillance videos. In this thesis, we take use of relevant methods and techniques of deep learning to complete the implementation, we improve the accuracy of entire recognition process by using these methods. In addition, the analysis of these methods and techniques can help us better understand the related processes of general human action recognition. We also improved the recognition methods by analyzing current deep learning human action recognition. Therefore, the research questions proposed in this thesis are:

(1) Which efficient deep learning methods can be applied to human behavior recognition?

(2) In order to realize human action recognition, YOLOv7 was selected as the basic model? How should we improve this method?

(3) *Will the use of attention mechanism in the model make human action recognition much efficient?*

(4) *After the experiment is successfully conducted, how to evaluate the model performance?*

The research questions raised are all progressive, which can help us better understand them. Human action recognition is the core of this thesis. To this end, we firstly look for what human action recognition methods are currently being proposed. We discovered that YOLO-based human action recognition method not only is executed quickly but also eliminates the needs to find additional feature extraction methods. Therefore, we choose YOLOv7 as the base model to implement human action recognition, and improve the model to get a better outcome.

1.3 Contributions

In this thesis, we mainly conducted the implementation of human action recognition. By the end of this project, we are able to achieve:

(1) *Efficiently recognize human behavior from video.*

(2) *Seamlessly filling the gap of human action recognition using YOLOv7.*

(3) *The accuracy of human action recognition is improved by adding an attention mechanism to the model.*

(4) *Timely Increasing the attention mechanism so as to reduce the time required for the recognition.*

This thesis will also introduce existing human action recognition methods. These contents will be introduced in the second chapter.

In addition, in order to get a better training effect. In this thesis, we combine multiple datasets to create an action dataset for training that fits the model used in this thesis. The focus of this thesis is on two attention mechanisms, which improve the accuracy of human action recognition. This is covered in Chapter 3.

1.4 Objectives of This Thesis

In this thesis, we firstly present a brief introduction to the related methods of human action recognition. Then, we discuss and analyze the principles of these methods. The extensive discussion will help us determine the method used in this thesis.

In addition, in order to be able to recognize human actions in video data, we propose three models for human action recognition based on YOLOv7. Based on YOLOv7, we combine CBAM and Siam attention mechanisms to improve the final recognition accuracy.

Finally, in this thesis, we also analyze the experimental results. We compare the proposed new model with the base YOLOv7 model and compare their performance on human action recognition. The selected method is determined by comparison and other evaluations as well as its final performance.

1.5 Structure of This Thesis

The structure of the thesis is as follows:

We will discuss literature reviews in Chapter 2. The analysis and introduction to the existing human action recognition are introduced related to human action recognition. In addition, Chapter 2 will introduce the relevant knowledge of deep learning and how these methods are applied to the disparate processes of human action recognition.

The research method of this thesis will be discussed in detail in Chapter 3. In this chapter, we will not only introduce the specific experimental method, but also present the overall design and layout of the experiment. Finally, we also introduce the dataset and evaluation method used.

We will brief the methods in Chapter 3 in Chapter 4. We will present the results of the experimental results in the form of graphs. This can help us better present the results of our research. In addition, the limitations of the method are also introduced in this chapter.

In Chapter 5, we will analyze and discuss the experimental results and in Chapter 4.

Finally, in Chapter 6, we conclude the whole thesis and present our future work.

Chapter 2
Literature Review

The focus of this project is on human action recognition based on deep learning. Therefore, in this chapter, we will concentrate on the related methods of human action recognition.

2.1 Introduction

The applications of human action recognition are very wide. In the field of intelligent surveillance, related technologies will gradually develop into the foundation of identification (AminUllaha, KhanMuhammadb, UlHaqa, & WookBaik, 2019). This is closely related to the development of human life. In the future of human life, surveillance cameras will only increase (Wang, Yan, Kankanhalli, Jain, & Reinders, 2003). In addition, video data will only become more and more. As a result, the need to identify human behavior in videos will only proliferate. It is precisely because of this prospect that the corresponding identification technology has gradually become the key to the research field, attracting more and more attention from technicians and scholars. In view of this, in this thesis, we will conduct in-depth research on various existing human action recognition technologies.

As a red hotspot in the field of computer vision, human action recognition is a direction with high research value and broad application prospects (Yan, 2020). The existing research work on human action recognition is still at the stage of gradual development. At the same time, human action recognition is a research topic that has received much attention, and there have been a great deal of related research work on human action recognition in the past few decades. In this field, a slew of new algorithms or frameworks have been proposed, explored, and exploited. Various identification methods based on different technologies emerge in an endless stream. The most critical method is related to the recognition based on deep learning (Zheng, Yan, & Nand, 2018).

Traditional popular method in human action recognition was to collect the action trajectories in the video data, extract the corresponding features according to the collected trajectories, and then classify the extracted features to obtain the final result. One of the best performing methods is a method called improved dense trajectories (iDT) (Wang & Schmid, 2013) (Wang, Kläser, Schmid, & Liu, 2013). However, video-based human action recognition is still a huge challenge. The reason is that a video often contains timing information that a image does not have. Due to the added timing information, a larger number of computations are required. However, with the development of deep learning technology and massive advances in computer computing power, such as the widespread use of GPUs. there are more and more human action recognition and related research based on deep learning. The use of deep learning technology can well solve the problems caused by the increase in computational complexity. Therefore, in the current recognition, deep learning has become the mainstream (Wu, Sharma, & Blumenstein, 2017).

Under the premise of using deep learning for human action recognition, deep neural networks will have different effects on the recognition results (Lu & Yan, 202). Therefore, deep learning networks are often the key to measuring the pros and cons of the entire action recognition technology. In the rest of this chapter, the corresponding methods and literature using different networks for action recognition will be briefly introduced, such as CNN, RNN, attention mechanism, Transformer, YOLO (Le, Nguyen, & Yan, 2021), etc.

2.2 Human Action Recognition Based on Convolutional Neural Network

Simonyan et al. (Simonyan & Zisserman, 2014) proposed a CNN-based two-stream network. There are two different flows in this network, spatial flow, and temporal flow. Among them, the spatial stream processes RGB image data, and the temporal stream processes optical flow data. In the structure of the network, the spatial stream takes raw video frames as input to capture visual appearance information, such as goals, scenes, etc. Temporal flow takes as input a stack of optical flow images to capture motion information between video frames, such as the motion information of the target object. The two streams will be trained separately in the network, each consisting of a CNN and a final SoftMax. The two networks are fused by using SoftMax. Finally, SVM was employed for classification. Furthermore, in the experiments, a multi-task training method was taken into consideration to combine two different datasets. The initial training data was augmented in this way, so that the final results are great based on both datasets.

Because of this improvement, CNN-based methods achieved performance close to the state-of-the-art non-deep learning methods at the time. This result further illustrates that motion information is very important for video action recognition. However, it is also noted that learning temporal information directly from raw video frames is still a very difficult challenge for CNNs. It is necessary to represent the movement by other means. At the same time, because of the success of this structure, there have been many further studies based on the dual-stream network. This has greatly promoted the development of video action recognition.

In the two-stream network structure, because two networks exist at the same time. A stage is required to combine the results of the two networks to obtain the final prediction. This stage is often referred to as spatiotemporal fusion. Since the network (Simonyan & Zisserman, 2014) was fused after softmax, the network cannot learn the correspondence between pixels of temporal and spatial features. In response to this problem, Feichtenhofer et al. (Feichtenhofer, Pinz, & Zisserman, 2016) proposed a new fusion method in 2016, which put the fusion of spatial network and temporal network in the convolutional layer. In addition, in order to improve performance, the author also replaced the basic spatial and temporal networks with VGG-16 network. The final results of the experiments show that the fusion in the last convolutional layer can also achieve good fusion accuracy. Also, fusion in convolutional layers reduces the need for fully connected layers in the network. Compared with the original two-stream network, this improved network uses half the parameters of the original network.

The network structure related to the basic two-stream network is relatively shallow (Krizhevsky, Sutskever, & Hinton, 2017). As the network deepens, the accuracy of the neural network tends to increase. Therefore, deeper network structures in two-stream networks become a noteworthy part. Wang et al. (Wang, Wang, Xiong, & Qiao, 2015) adopted deeper temporal and spatial networks. Although deeper architectures achieved higher network performance in space, deeper temporal networks do not yield better accuracy by simply using a deeper network does not yield better results. After sufficient analysis, they believe that this is due to the video dataset is too small, resulting in an overfitting problem in the final result (Soomro, Zamir, & Shah, 2012) , (Kuehne, Jhuang, Garrote, Poggio, & Serre, 2011).

In order to solve the problem of overfitting, Wang et al. took use of data augmentation methods: Corner cropping and multi-scale cropping based on the old network. It also reduces the learning rate of the deep network and enhances the pre-training step. At the same time, the dropout ratio is increased. More GPUs are utilized to train the network. Through these methods, overfitting of deeper networks was successfully prevented. Final experimental results demonstrated that the new deep network achieves a recognition accuracy of 91.4% (Wang, Xiong, & Qiao, 2015).

In fact, one of the drawbacks of the two-stream method is that it can only collect short-term video motion information, and it is somewhat powerless for the motion information in those long-term videos. To solve this issue, Wang et al. proposed a Temporal Segmentation Network (TSN) for video-level action recognition (Wang, et al, 2016). In this network, a new idea was put forward, which is to split a long video into several short segments that can be feature extracted by a two-stream method. Each segment will give its own spatiotemporal features for the behavior category, and finally these features will be combined to achieve the final prediction result. They also made changes to the part of the two-stream network in the text and made sure they were the best. For example, in terms of the type of data, two inputs, RGB difference and warped optical flow field, were tried. The final result is that the combination of RGB+Optical Flow+Warped optical flow works best. In terms of structure selection, three network structures, GoogLeNet, VGGNet-16 and BN-Inception, were tried. The experimental results show that BN-Inception works best. Furthermore, they demonstrated that using TSN networks to train very deep networks on a limited training set can avoid severe overfitting. The final experimental results show that the network has excellent performance, achieving a good result of 94.2% on the UCF101 dataset.

Because TSN has very superior performance and is relatively simple to create. Therefore, most of the latter two-stream methods were later developed based on TSN.

The improved network of TSN also emerges in an endless stream. Lan et al. proposed an improvement on the fusion part based on TSN. They approach a deep neural network as a local feature extractor. After extracting multiple local features, these local features are aggregated into global features. Finally, the result is obtained by SVM classification. The experimental results show that the final effect of this method is significantly improved.

Zhou et al. (Lan, Zhu, Hauptmann, & Newsam, 2017) improved on temporal relational reasoning. A novel network based on TSN, namely, Temporal Relational Network, was proposed. The network is structured to perform temporal reasoning on the input feature maps. Different features are obtained by temporal reasoning on video frames of different lengths. Finally, these features are fused to get the final result. In addition, three fully connected layers are added to learn the weights of video frames of different lengths. The final experimental results demonstrate that the proposed TRN network gives the network the ability to discover temporal relationships in videos, and the results are excellent.

Based on the two-stream convolutional network, Zhu et al (Zhou, Andonian, Oliva, & Torralba, 2018) applied pooling to space-time and proposed a deep learning network called Temporal Pyramid Pooling (DTPP). The network samples the input of the two streams respectively to obtain the corresponding temporal and spatial features. These features are then combined for pre-training using a temporal pyramid pooling layer. The resulting model not only has multiple time scales, but is also globally and sequence aware. In experimental design, DTPP will first be pre-trained on ImageNet or Kinetics. Then the test experiment will be carried out. The final experimental results show that the performance of the model is good enough.

2.2.2 3D CNN Network

The application of 3DCNN in human action recognition was proposed. (Ji, Xu, Yang, & Yu, 2012). CNNs at the time could only handle 2D data. But video data is often 3D related, in order to recognize human actions in the video, they created a 3D-CNN model. The model can directly extract features with spatial and temporal relationships through a special 3D convolution kernel in the network. The feature is directly extracted from the input video, and the motion information of multiple adjacent frames is obtained through the convolution kernel, the final result is obtained according to this information. After many experiments, it has been proved that the use of 3D-CNN model for video human action recognition has a good effect compared with other methods at that time.

To further exploit the potential of 3D CNNs, Tran et al. (Tran, Bourdev, Fergus, Torresani, & Paluri, 2015) extended 3D CNN from shallow to deep networks, called C3D. Because the model directly uses deep 3D convolutional networks (3D ConvNets) to process spatiotemporal features. Therefore, in terms of learning spatial features, this method was much faster than the Two-Stream method, and the training process was basically end-to-end training, and the network structure is more concise. The final test results show that the model has broad applicability and is easy to train and use.

But C3D also has some problems. Due to the depth problem of the whole network, increasing the depth inevitably enhances the number of network parameters. This makes training a C3D take longer. To solve this problem, Sun et al. (Sun, Jia, Yeung, & Sh, 2015) , proposed the FstCN network structure in 2015. They made improvements to the 3D filters in the network. Decompose the 3D filter into a 2D and a 1D filter combination. This reduces the parameters of the network. The final experimental results prove that the dataset trained by this model achieves good performance in results. However, due to other problems of C3D, the mainstream network in this period is still a two-stream network dominated by 2D CNN.

The advent of I3D brought a turning point for this approach. The I3D method was proposed by Carira et al. (Carreira & Zisserman, 2017) and achieved a height of 95.6% on the UCF101 dataset. Structurally, I3D will be based on the inception-V1 model, extending from 2D to 3D. In addition, I3D has also successfully used the image classification architecture in 3D CNN. By conducting and training on Kinetics400, the problem that 3D CNN network needs to be trained from scratch every time during training is solved. The emergence of this method provides a successful idea for other 3D CNN methods.

Diba et al (Diba, et al., 2017) proposed a new network structure Temporal 3D ConvNets (T3D) based on 3D convolution. Structurally, T3D is implemented based on DenseNet. The 2D convolution kernels are replaced with 3D convolution kernels. Furthermore, they propose a new structure called Temporal Transition Layer to replace the Transition layers in DenseNet. In addition, in order to solve the problem of difficult training of 3D convolutional networks. The authors successfully transfer the pre-trained weights in the 2D CNN to the 3D CNN using the transfer learning method. With the above improvements, their network can be directly trained on some smaller datasets and achieve better performance of the initial 3D CNN network. And the final performance of the model trained by this network is better than other methods at the time.

The same is for improving 3D convolutional networks. Qiu et al (Qiu, Yao, & Mei, 2017). proposed a network structure called P3D. In terms of network structure, the author replaces the convolution kernel in ResNet Units with a 3D convolution kernel on the basis of ResNet. Unlike other methods, they are not directly replaced by 3D convolutions. Instead, 3D convolution is approximated by a combination of spatial and temporal convolutions instead of 3D convolutions. Through this method, not only the number of model training parameters is reduced, but also the advantages of 2DCNN in pre-training are brought into play. Finally, the authors finally demonstrate the excellent performance of the model on different tasks.

Shou et al. (Shou, Chan, Zareian, Miyazawa, & Chang, 2017) combined convolution and de-convolution into the network based on C3D, and created a network called Convolutional-De-Convolutional (CDC) Networks. The network structure can not only upsample in the temporal dimension, but also downsample in the spatial dimension at the same time. In addition, the authors believe that the granularity of the methods used in the past is not high enough, and for this they believe that positioning using temporal actions can effectively improve this value. In the final experimental results, the method runs fast and improves mAP. This is enough to prove that the method is very effective.

Although the structure of the CDC model is cleverly designed, there are also some problems. In a CDC network, there is a problem of loss of timing information when collecting information. To solve this problem, Yang et al. (Yang, Qiao, Li, Lv, & Dou, 2018) proposed an improved CDC model and named it Temporal Preservation Networks (TPN). In this model, the ordinary temporal convolution in the network is replaced with temporal preservation convolution. This ensures that the network will keep the size of the receptive field unchanged without performing the timing pooling operation. And this change will not shorten the timing length in the network, so that the timing information of the network is better preserved.

Furthermore, due to the large number of parameters in deep convolutional neural networks. Overfitting often occurs as a result. To change the appearance of this result, using a regularization method is a good option, Kim et al. (Jinhyung Kim, Wee, Bae, & Kim, 2020) proposed a simple regularization method to avoid this problem. This method is called Random Mean Scaling (RMS). The key to this technique is to regularize the model by changing the magnitude of the low frequency components of the features by RMS. And RMS can enhance the entire model by adding only a small amount of computation during training. After extensive experiments, the results show that the method is very effective compared to other state-of-the-art regularization methods.

After a summary of the above literature, we find that the related research on the use of CNN for human action recognition has been relatively mature and has different methods. Therefore, we will introduce other deep learning networks and methods for human action recognition in the rest of this chapter.

2.3 Human Action Recognition Based on Recurrent Neural Network

In order to obtain better results, a method that fuses LSTM with a two-stream network is proposed (Ng, et al., 2015). In this method, an LSTM is used as the fusion part of the two-stream network to connect with the underlying CNN. The final result of the experiment proves that the final effect of this improvement is comparable to other methods.

Donahue et al. is combining 2D CNN and LSTM network (Donahue, et al., 2015). A model called Long Term Recurrent Convolutional Network (LRCN) was proposed. Structurally, the basis of the CNN structure of LRCN is CaffeNet. The network extracts the features of each frame by using a 2D CNN, and then uses an LSTM to generate the corresponding action labels. Through this simple process, actions are identified through the network.

The two articles both firstly combined LSTM with the two-stream network environment to a certain extent for video action recognition. They took use of the features obtained by CNN in the network as the input of the deep LSTM network and aggregate these features from frames into video results. In the two-stream network, they incorporated the temporal stream and the spatial stream into the LSTM respectively, but the final result is still the final output by fusing the results of the two streams. In the experimental results, despite this has achieved good results, adding the LSTM model did not produce a large improvement in results compared to the original and two-stream baseline networks. On the basis of this framework, more methods using LSTM have been proposed.

Based on this framework, more methods using LSTM have been proposed. Li et al. (Li, et al., 2016) achieved action recognition by creating a hierarchical multi-granularity LSTM network. The network was designed based on end-to-end way. Therefore, it has a good performance in the experimental results. In addition, the author also proposes a new network fusion scheme based on multi-granularity score distribution.

Gammulle et al. (Gammulle, Denman, Sridharan, & Fookes, 2017) proposed a framework for deep synthesis of CNN and LSTM. This framework maps the spatial features obtained through the CNN network to temporal relations through the LSTM network. This enables the framework to effectively utilize both spatial and temporal features in the network. In addition, the two sets of features, which are completely combined, act as the attention mechanism of the network to a certain extent. And this framework also achieves high accuracy in results.

Ullah et al. (Ullah, Ahmad, Muhammad, Sajjad, & Baik, 2017) have also made efforts in this regard. They also use the structure of CNN plus LSTM for human action recognition. But instead of using the common LSTM structure, they used a network called deep bidirectional LSTM. In the identification method, the method uses a feature extraction method that can reduce the complexity of the network. Then, the extracted deep features are put into the DB-LSTM network to increase network in depth through bidirectional propagation. In this way, the model can identify and learn those long-term sequences. After experiments, the identification method of this design has a significant improvement compared with other methods of its time.

2.3.2 Other RNN Method

Because the simple application of RNN network to the recognition of video sequences does not play a very good role, the networks can even become ineffective when the duration of the action to be recognized becomes longer. To solve this problem, a network called Lattice-LSTM (L2STM) was proposed (Sun, et al., 2017). In order to accurately simulate long-term complex motion, the network extends LSTM. The specific idea of this extension learns the process of changing the state of a memory cell at a single spatial location. Furthermore, this shift does not increase the complexity of the model. To make the training of the model better, they also propose a new joint training method. These improvements enabled the L2STM network to perform well on two fixed datasets and outperformed other well-known recognition methods at the time.

Although RNNs are the crux of the current sequence learning problem, there are still many problems in concrete practice. Shi et al. (Shi, Tian, Wang, Zeng, & Huang, 2017) suggested that this may be related to the lack of feedback connections in the structure of the network. Therefore, a structure called ShuttleNet was proposed by using GRU instead of RNN as the base processor of the model. For the network to perform an analogous function as the feedback connection, the internal processors of the network are all cyclically connected.

With this improvement, the existing network becomes a parallel work. It conducts this performance through feedforward and feedback connections in the network. In the final experiments, they got good results after adding this structure to the RNN network framework. Through this improved experimental data, we can also see that using GRU in the network requires fewer parameters than simply using RNN. And on the action recognition task, we get analogous performance to the network using LSTM.

Zheng et al. (Zheng, An, & Ruan, 2017) proposed a multilayer RNN network structure. They named this network structure as Multi-Level Recurrent Residual Networks (MRRN). There are three different levels of recognition flows in this network, namely low, medium, and high recognition flows. The basic structure of each recognition flow is composed of a ResNets network and a recurrent network model. The model fuses three streams that learn independently through a weighted average. In addition, since the model also reduces the time and space complexity of the network through shortcut connections. This enables the network to be trained end-to-end with higher efficiency. In the experimental results, the performance of the network has produced obvious improvement, and it is not bad compared with the new technology at that time.

Most neural networks that are use of RNNs for human action recognition were designed to rely on a two-stream architecture. However, the existing two-stream RNN network (Sun, et al., 2018) does not fully utilize the information in the network. Based on this idea, the idea of exploiting the information was proposed in the network in a circular fashion. So, a novel coupled recurrent network (CRN) was propounded. The most important thing in this loop structure is the module called Loop Interpretation Block (RIB). Through this module, multiple inputs to the network are processed and the features are extracted. In addition, in order to improve the training performance of the model, they also took use of an efficient training strategy suitable for the CRN network. Experimented on this basis. Ultimately, they demonstrate the effectiveness of the CRN network and achieve state-of-the-art results.

2.4 Attention Mechanism

In deep learning, most design concepts are inspired by the human biological nervous system. Among these concepts, the attention mechanism can be regarded as a particularly important concept. Today's attention mechanism is one of the most unique of deep learning-related concepts. Using attention to process large amounts of information, it focuses on those unique parts of the information.

The concept of attention was firstly proposed from Google DeepMind. (Bahdanau, Cho, & Bengio, 2015). The attention was applied to create a network for machine translation of people, called RNNsearch. RNNsearch is structurally divided into an encoder and a decoder, both of which are composed using a bidirectional recurrent neural network (BiRNN). Later, with the further development of deep neural networks, the attention mechanism has been widely used in various application fields. It is also used in the field of human action recognition.

Based on CNN and LSTM networks, Sharma et al. (Shikhar Sharma, 2015) pioneered adding an attention mechanism to the network structure. By adding an attention mechanism to the network, the network can only focus on the part of the data set that is strongly related to the behavior category. In order to prove that the introduction of attention mechanism makes the network better, the author compares two different networks, which further proves that this attention CNN-LSTM network is better than the traditional LSTM.

Du et al. also added an attention mechanism to the network. They propose a mechanism called pose-attention. Adding attention to the network not only allows the RNN model to learn more complex motion structures over time, it also enables simple pose labeling of videos through the network. The proposal of this pose attention network enriches the field of action recognition using RNN to a certain extent.

A network called VideoLSTM was proposed. In order to apply ConvLSTM to human action recognition, an attention mechanism based on the network was introduced. Finally, it got the VideoLSTM network. A new attention map is applied in this network structure. In addition, the network explains how to use attention on action localization via action class labels. Finally, the network shows improved results for action recognition on the results.

The end-to-end model is important part. Song et al. proposed another recognition network with an attention mechanism based on LSTM (Du, Wang, & Qiao, 2017). The whole network consists of three parts, namely, the main LSTM network, the spatial attention subnet, and the temporal attention subnet. Among them, the network learns the relationship between nodes in disparate frames through LSTM. Based on this relationship, the spatial and temporal attention subnets find out which data in the data has the greatest impact and contribution to action recognition. The importance is automatically assigned to disparate joint points through the content of the sequence, that is, disparate attention is given. The assignment of joint importance tends to change over time. In addition, the authors employ an alternate joint training approach to train the network and design a regularized loss function to prevent the model from overfitting.

The networks are based on LSTMs combined with attention mechanisms. This also means, these networks are implemented based on RNN. However, the attention mechanism is not only realized based on RNN, and a structure that realizes attention without the RNN network appears.

Chen et al. (Yunpeng Chen, 2018) proposed a dual attention network (A2-Nets). As the name suggests, this is a network consisting of two attention modules. An attention module is responsible for collecting all spatial features together to generate a global feature set. The second module is responsible for distributing these features to suitable locations. It takes advantage of two attention maps to collect and distribute long-range features. In addition, it is also very convenient and fast to add this network to the existing deep neural network. The performance of this model was evaluated by conducting experiments on a video recognition dataset. On the action recognition task, the model achieved better performance than other models at the time.

The attention mechanism was combined to the CNN framework and named as the framework AssembleNet++ (Ryoo, Piergiovanni, Kangaspunta, & Angelova, 2020). A network component called peer-attention was put forward. Through this component network, the importance of spatiotemporal features of different convolutions can be learned, and different weights can be assigned to them. In addition, the attention mechanism can also be widely used in other CNN models. The final results of the method showed that the method outperformed other methods at the time.

Transformer is a model implemented entirely based on attention mechanism. This model completely abandons convolution and recursion, and adopts an encoder-decoder structure in structure. It was first proposed by Vaswani et al. (Vaswani, et al., 2017) for translation tasks of text data and achieved good results. After the concept of Transformer was proposed, the model achieved great success in natural language processing. In the task of dealing with long-term time series modeling, it has shown much better performance. Therefore, the applications of transformers to human action recognition tasks have become the research direction of many people.

Bertasius et al. proposed a new video action classification method (Bertasius, Wang, & Torresani, 2021), TimeSformer, based on the structure of Transformer. By decomposing video into frame-level patches, this method enables direct spatiotemporal feature learning. Through such improvements, the standard Transformer can be employed in video action classification. In addition, they also applied spatial and temporal attention to each block of the modulo model respectively, and called this design divided attention. The test results show that the newly designed TimeSformer has achieved excellent results in the test.

Video Transformer Network (VTN) is a network structure for action recognition. It takes use of ViT to model space and an attention-based encoder to model time (Neimark et al., 2021). Among them, ViT was often used as a Transformer in image classification tasks. In the network, the spatial features were extracted from the transformer and input into the attention-based encoder, MLP was taken to output the final result. The experiments show that the network gives excellent results on the task of video human action recognition and runs faster than other methods (Neimark, Bar, Zohar, & Asselmann, 2021).

Fan et al. applied Transformer to human action recognition in a clever way. Because Transformer has high classification performance. So, 3D frames of the video were converted into a 2D super image that can be directly input. The transformed data will go through a Transformer-based image classifier to complete action recognition. The experimental results proved that their idea was correct. The recognition results using this method on the Kinetics400 dataset achieve analogous performance to some of the best CNN methods (Fan, Chen, & Panda, 2021).

ViViT is a video human action recognition Transformer proposed by Arnab et al. They took use of four different spatiotemporal Transformer combinatorial structures to address long sequences of tokens appearing in videos. Furthermore, they also effectively regularized the model during training. After the extensive experiments, it is proved that the Transformer performs well in action recognition (Arnab, et al., 2021).

A model called Action Transformer (AcT) was proposed. The appearance of this model was inspired by Vision Transformer which is a Transformer model based entirely on self-attention. Furthermore, in order to get an accurate model, a new dataset MPOSE2021 was obtained. After extensive testing on the dataset, the proposed model consistently outperforms other networks using CNNs and RNNs for recognition (Mazzia, Angarano, Salvetti, Angelini, & Chiabergea, 2022).

In order to reduce the GPU memory required in the recognition process, a novel framework called Recurrent Visual Transformer (RViT) was proposed (Messina, Amato, Carrara, Gennaro, & Falchi, 2022). The framework reduces GPU memory requirements by using frame-by-frame processing. In RviT, the attention gate structure was added. This structure allows the network to establish an interaction that associates the input frame with the previous hidden state. RViT also incorporates the concept of circular execution in the network. RViT takes use of both to obtain spatial and temporal features in videos.

The final experimental results prove the superiority of RViT, and RViT always performs well on different datasets. For example, RViT can achieve 92.31% on the Jester dataset.

These methods all reduce the complexity of Transformer's understanding model to a certain extent. At the same time, the calculation amount of using Transformer is reduced to a certain extent. In addition, it is equally feasible to introduce Transformer into neural network to help human action recognition.

The current CNN-based action recognition still has some limitations. Affected by the receptive field, the existing CNN models cannot capture long-range temporal information. Therefore, Hussain introduced Transformer. Specifically, after extracting features using the Vision Transformer, the features are passed into a multi-layer LSTM. Finally, the long-term dependencies of the action were obtained. The addition of Transformer not only helps the network obtain relevant information about remote time, but also encodes spatial information relative to temporal information. Through extensive experimental experiments on UCF50 and HMDB51, they finally determined that the model improved the accuracy by 0.944% and 1.414% (Hussain, Hussain, Ullah, & Baik, 2022).

2.5 Other Multiple Deep Learning Method for Human Action Recognition

Using other methods in combination with deep learning methods can get satisfactory results for human action recognition. Applying trajectories to neural networks is an option (Yan & Kankanhalli, 2003). Wang et al. proposed a deep learning method using trajectories, Trajectory Pooling Depthwise Convolutional Descriptors (TDD). The features obtained using this method consider the advantages of traditional handcrafted features and deep learning features. The learned convolutional features are integrated together by trajectory constraints. These features are then turned into valid descriptors by using two different normalization methods. It is worth noting that the trajectory hereinafter refers to the path that the pixel is traced in the temporal dimension. To determine the effectiveness of the method, the experiments were conducted by using the

UCF101 and HMDB51 datasets. The experimental results show that the method performs well and is one of several methods that perform best on both datasets at the time (Wang, Qiao, & Tang, 2015).

It is helpful to extract video features by improving the recognition system using data-driven and data-independent methods. Therefore, IDT was incorporated into the network while a two-stream stacked convolutional independent subspace analysis (ConvISA) architecture was proposed. Through this method, the new local feature descriptors are perfectly combined with the hand-crafted descriptors for subsequent classification. After experimenting with four datasets, the method reached the state-of-the-art at the time (Lan, Yu, Lin, Raj, & Hauptmann, 2015).

In an action video, the number of frames where action can be discriminated may only be in a few key frames. A general action recognition network assigns labels to all frames. Zhu et al. proposed a deep learning method to identify key volumes. The network looks for the Key Volume for each action of the input. And use the found data to update the parameters of the network. This model finally achieved 93.1% accuracy on the UCF101 dataset (Zhu, Hu, Sun, Cao, & Qiao, 2016)

Diba et al. proposed an encoding layer called Temporal Linear Encoding Layer (TLE) in order to fuse and encode people in different positions in the video. When this encoding layer is added to the 2D CNN, the network can capture all appearance and motion throughout the video. Moreover, using this network to model data is not easy to lose information and the modeling method is more expressive. Furthermore, this encoding layer can also be used in 3D CNN. In feature extraction, the author used two different networks, Two-stream and C3D, to extract features. Experiments show that the recognition effect of TLE is very good (Ali Diba, 2017)

Zhao et al. proposed a new convolution algorithm based on trajectory, named as TrajectoryNet. In the basic structure of the network, a two-stream convolutional network is chosen. TrajectoryNet was proposed for model training from the data in the temporal dimension instead of conventional temporal convolution. Spatial convolution continues to adopt regular convolutions. This approach without CNN as a fixed feature extractor performs well on the Action Recognition dataset dataset, with a significant improvement over the base two-stream network (Zhao, Xiong, & Lin, 2018).

Yucer et al. proposed a different system for 3D human action recognition. The system consists of two modules. The first module consists of the Siamese-LSTM network to form the basic function. Throughout this module, a similarity measure between two 3D joint sequences in the network can be learned. The second module is responsible for the final identification of the output of the first module. It is worth noting that the first module can be separated from the system and trained independently. This can also get the final recognition result. Although the final experimental results are not excellent, the overall idea is worth learning. This system can be further developed by adding LSTMs in the future (Yucer & Akgul, 2018).

The results of human action recognition are often affected by environmental factors such as background clutter and illumination changes. In order to understand which network structure can avoid the influence of these factors, Yu et al. implemented three different action recognition models: 3D-CNN, Two-Stream network, and CNN+LSTM. After extensive experiments by placing these models on the HMDB-51 dataset. They found that only the CNN+LSTM model can effectively avoid the influence of interference factors (Yu & Yan, 2020).

Liang et al. also chose CNN+LSTM network as the method of human action recognition. By comparing the four structures of KNN, KNN+STIP, CNN, CNN+LSTM, they finally determined that using CNN+LSTM network for action recognition can get a good accuracy (LIANG, LU, & YAN, 2022).

2.6 YOLO for Human Action Recognition

Among the target detection methods based on deep learning, we divide them into two categories according to the detection method (Jiao, et al., 2019) : Two-stage detection and one-stage detection. Most of the methods at that time utilized deep learning methods to analyze pictures, find out the areas where objects exist, and cut these areas out. The deep learning human action recognition methods, such as CNN, etc. Most of these methods are two-stage detection methods. This kind of methods are also called the two-stage methods. In general, the localization and target recognition accuracy of two-stage detection will be relatively high. The detection speed of single-stage detection is faster (Jiao, et al., 2019).

One-stage detection is to directly classify each region of interest as a background or target object (Wu, Sahoo, & C.H.Hoia, 2020). The detection method can directly give the class probability of the object through only one stage. The most typical representative of this is YOLO (You Only Look Once) (Redmon, Divvala, Girshick, & Farhadi, 2016).

YOLO is an advanced one-stage object detection framework. It has now gone through 7 versions of the evolution. In addition, YOLOR (Wang, Yeh, & Liao) which combines traditional compressed sensing and YOLOX (Ge, Liu, Wang, Li, & Sun, 2021) which does not rely on anchor boxes are also produced. As shown in Figure 2.1, we can briefly view the development process of the YOLO detection model through this figure.

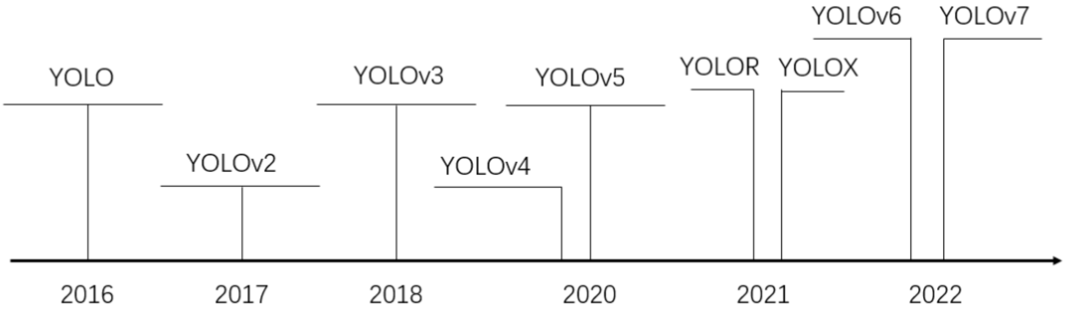


Figure 2.1: The process of YOLO model

YOLOv1 in 2016 directly divided the image into several regions, and simultaneously predicted the bounding box and probability of each region, and the detection speed was greatly improved. However, the shortcomings are also obvious. Compared with the two-stage detector at that time, the positioning accuracy is lacking, especially for the positioning of small targets (Redmon, Divvala, Girshick, & Farhadi, 2016).

The basic framework of YOLOv1 is shown in Figure 2.2. Firstly, we adjust the input image size to 448×448, and send it to CNN to extract features, and then tackle the network prediction results to achieve end-to-end target detection. Structurally, YOLOv1 took use of a backbone network like GoogleNet (Szegedy, et al., 2015) with 24 convolutional layers and 2 fully connected layers. It is pre-trained on ImageNet and then transferred to the detection task for validation on the VOC dataset (Everingham, et al., 2015). In addition, YOLOv1 segments the input image into 7×7 grids, and each grid predicts two bounding boxes, so there is 7×7×2 bounding boxes. Identify up to 49 targets. Therefore, YOLOv1 is not conducive to identifying dense objects and small objects.

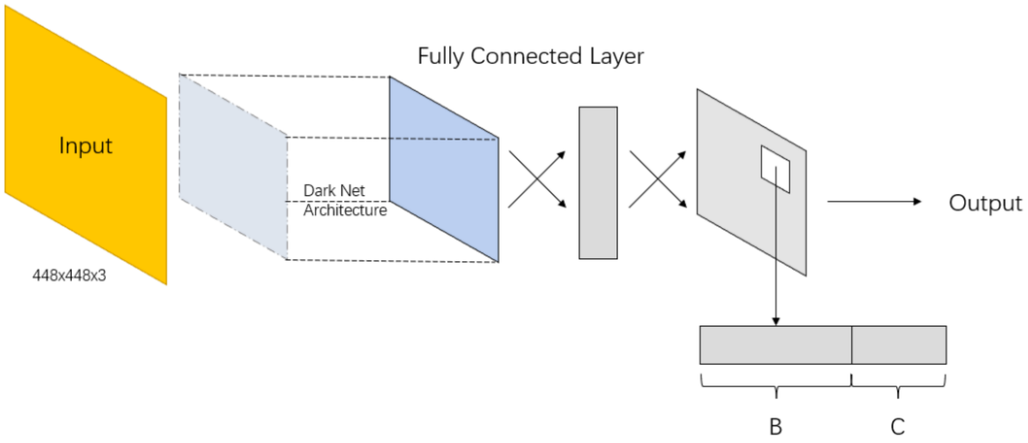


Figure 2.2: The basic framework of YOLOv1 model

YOLOv1 abandons the traditional sliding window technology. Its CNN divides the input image into an S×S grid, and then each cell is responsible for detecting those targets whose center points fall within the grid, and each cell predicts B boundaries box and bounding box confidence. Confidence contains the probability that the bounding box contains the target and the accuracy of the bounding box. Each bounding box predicts 5 elements: (x, y, w, h, c), which represent the position, size, and confidence of the bounding box, respectively. Each cell predicts $(B×5+C)$ values, where C is the number of categories. Afterwards, the Non-Maximum Suppression (NMS) algorithm is used for network prediction.

These problems affect the recognition performance to some extent. So, in order to solve these problems. Based on YOLOv1, YOLOv2 was proposed (Redmon & Farhadi, 2017). Compared with the previous YOLOv1 version, while maintaining the processing speed, YOLOv2 has made improvements in three aspects: More accurate prediction, faster speed, and more recognized objects. YOLOv2 draws on the VGG network to build a new backbone network Darknet-19. Because YOLOv1 took use of the fully connected layer to directly predict the bounding box, it loses a lot of spatial information, resulting in inaccurate positioning. Therefore, YOLOv2 introduces anchor boxes to replace the fully connected layers of v1 to predict bounding boxes. YOLOv2 pioneered a training method that uses a combination of classification and detection to extend object detection to objects lacking detection samples. Significantly improve the prediction accuracy while maintaining the advantage of fast inference.

Specifically, YOLO9000 can be regarded as an extension of YOLOv2. It has made the following improvements based on YOLOv2, which greatly improves the detection accuracy:

(1) YOLOv2 took use of a basic network designed by itself, and the network is designed with consideration. The amount of calculation of convolution makes YOLOv2 faster. In YOLO9000, batch normalization is added to the base network to make the network converge faster.

(2) Before training the detection network, it needs to fine-tune the pre-trained classifier on the high-resolution pictures to make the network adapt to the resolution of the detected pictures in advance. It also improves the accuracy of the classification network to a certain extent and obtains a better classifier.

(3) The convolution was employed to replace the full connection of YOLOv2 for the parameters of the regression target.

(4) By returning to the width and height of the target, the multiscale reference frame is used for matching training to reduce the positioning error of the detection.

(5) Detection not only needs to classify the target, but also needs to locate the target.

The classification needs high-level semantic features, and the positioning needs the detailed information of the picture. In this method, the cross-layer feature fusion is used to obtain multi-scale features, and the result is the convolutional features of can be well suited for detection.

The basic network of YOLOv3 is Darknet-53, which draws on the residual structure of ResNet (He, Zhang, Ren, & Sun, 2016) to deepen the network structure while preventing the problem of network convergence caused by network gradient explosion. During the forward pass, the pooling layer and the fully connected layer are removed, and the size of the tensor is changed by changing the stride of the convolution kernel. Like v2, Darknet-53 will reduce the output features to 1/32 of the input, so the input image resolution is usually required to be a multiple of 32 (Redmon & Farhadi, 2018).

At the same time, YOLOv3 utilizes tensor splicing to expand the dimension of the tensor to extract more information. The specific operation is to splice the Darknet-53 middle layer and a subsequent layer after upsampling. Without affecting the detection speed, the accuracy of YOLOv3 is increased by about 1 percentage point, and the convergence speed is faster, which further improves the target detection ability of YOLOv3.

Darknet-53 has a total of 53 convolutional layers from layers 0 to 74, and the rest are residual layers. The 75th to 105th layers are the feature fusion layers of YOLOv3, in which YOLOv3 adds multi-scale detection (equivalent to neck), using 3 scales, and the outputs are 52×52, 26×26, 13×13 are used for detection small, medium, and large targets, each scale predicts 3 anchor boxes.

In short, the predicted frames of YOLOv3 are more than 10 times larger than those of YOLOv2, and they are performed at different scales, so the overall detection accuracy and the detection accuracy of small objects have been greatly improved, so YOLOv3 can be regarded as a single stage one of the milestone algorithms of target detection.

YOLOv4 summarizes various improvement methods after YOLOv3. Meanwhile, YOLOv4 is more suitable for training on a single graphics card (Bochkovskiy, Wang, & Liao, 2020). YOLOv4 is mainly divided into two modules, one is a module that improves training without affecting the inference speed. The other is a module with less impact on inference time and higher performance reward. For example, the local Cross Stage Partial (CSP) (Wang, Chen, Hsieh, & Yeh, 2020) adopted in the backbone network maintains high inference speed while still having high accuracy.

YOLOv4 selects CSPDarknet-53 as the backbone network. This is because Bochkovskiy et al. found that when the model is optimal for classification, its detection may not be optimal. The classification accuracy of CSPResNeXt-50 is higher than that of CSPDarknet-53. However, the detection accuracy of the latter is higher than that of the former. Therefore, CSPDarknet-53 is more suitable as the network backbone (Bochkovskiy, Wang, & Liao, 2020).

In the overall structure of this network, the overall architecture of YOLOv4 is the same as that of YOLOv3, but each substructure has been improved. Structurally, YOLOv4 deletes the last pooling layer, fully connected layer and softmax layer. Its backbone network consists of 5 CSP modules. YOLOv4 introduces Spatial Pyramid Pooling (SPP) and Path Aggregation Network (PANet) modules into the neck module. Increasing the receptive field via SPP separates important contextual features without slowing down the runtime. PANet was adopted instead of Feature Pyramid Network (FPN) in YOLOv3 for parameter aggregation, and use tensor connections to replace the original short connections. The head module, YOLOv4 has not made too many changes, and still inherits the multi-scale idea of YOLOv3 for prediction. YOLOv4 performs recognition better than YOLOv3 overall. YOLOv4 has a greater mAP and a faster training speed in terms of results. Additionally, YOLOv4 performs better than YOLOv3 at obscured object recognition. The structure of YOLO can also be further improved (Gai, Chen, & Yuan, 2021).

The basic structure of YOLOv5 (Wu, et al., 2021) is like YOLOv4, but it is scaled according to the scale of different channels, and five models of YOLOv5-N/S/M/L/X are constructed from small to large according to the model. The detect speed of YOLOv5 is significantly faster than that of YOLOv4, but the detection performance is not significantly different from that of YOLOv4. Additionally, YOLOv5 makes it easier to train its own dataset.

YOLOX is based on the basic structure of YOLOv3 (Bochkovskiy, Wang, & Liao, 2020) and YOLOv5 (Wu, et al., 2021). CSPNet, SiLU activation functions and PANet are used. In addition, YOLOv4 models of YOLOX-S/M/L/X have been designed. These models are all scaled by YOLOX. In addition, the model is further reduced to build YOLOX-Tiny and YOLOX-Nano for mobile edge devices.

However, methods such as YOLOv5 and YOLOX still have a lot of room for improvement in efficiency and speed. YOLOv6 and YOLOv7 are proposed on this basis. YOLOv6 has two types of models in structure. The large model uses the CSPStackRep block as the base block, and improves the PAN neck and proposes the Rep-PAN neck. Small models use RepBlock as the base block. In addition, YOLOv6 also borrows the structure of the YOLOX decoupling head, which has achieved good results in both classification and regression tasks (Chuyi Li, Chu, Wei, & Wei, 2022).

YOLOv7 follows YOLOv6 and was proposed by the author team of YOLOv4. Under the same volume, YOLOv7 is more accurate than YOLOv5, 120% faster (fps) and 180% faster than YOLOX (fps). YOLOv7 uses an extended efficient layer aggregation network to enhance the ability of the network to learn. A model scaling method based on concatenate model is proposed, which can keep the characteristics of the model in the initial design and maintain the best structure (Wang, Bochkovskiy, & Liao, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, 2022). Also, tag matching has been reset. The specific structure of YOLOv7 will be introduced in detail in Chapter 3.

Park et al. propose (Park, Park, & Kim, 2018) a novel object segmentation scheme to improve human action recognition. They use the YOLO Action Network's object detector. Object location and number of objects detection done using the YOLO network.

faster than other methods. Additionally, YOLO can instantly produce item classifications and associated probabilities from full photos. Experimental results show that the accuracy of the new scheme is improved by 15%. Wu et al. used YOLO in human action recognition based on skeleton data. The YOLO network they designed has only 1 dimension. In the end, their mAP reached 81.88% (Wu, et al., 2019).

In order to complete human action recognition, Lu et al. conducted related research work using the YOLOv3 model. To maximize the accuracy of their recognition, they were experimenting with public datasets, and they also created another dataset for their experiments by collecting their own data. In addition, in order to obtain a high-precision network learning rate suitable for the network, the network structure is continuously adjusted. In addition, the results of YOLOv3 are compared with the results of YOLOv2. This makes the method more credible and the experimental accuracy finally reaches 80.20% (Lu, Yan, & Nguyen, Human Behaviour Recognition Using Deep Learning, 2018).

In 2020, Lu et al., YOLO network for human action recognition is further optimized. They proceeded to human action recognition using more advanced methods than YOLOv3. They used the more advanced YOLOv4 combined with LSTM network. Furthermore, they further improved the performance of the network by adding event and spatial information to the recognition method, as well as an attention mechanism. After a lot of experiments, the accuracy rate reached 97.87%. To further improve network performance, a new Selective Kernel Network (SKNet) model with attention mechanism was added to their mailbox network. Finally, an accuracy of 98.70% is achieved (Lu, Yan, Nguyen, 2020).

By reviewing the various methods mentioned in the literature review, we see that good results can be achieved on action recognition using a variety of different methods. Among them, the YOLO model has great potential. By adding other structures to the YOLO model, the final recognition effect can often be greatly improved (Luo, Yan, & Nguyen, 2022).

By sorting out related work, we found that the methods of human action recognition using CNN or RNN often need to join multiple networks to improve the effectiveness of the experiment. Also, since the methods implemented using these methods are often not end-to-end, they are not very fast. The YOLO series of methods run fast and are easy to train. Additionally, few people are using YOLO for human action detection research. The method used is also YOLO or YOLOv5, and the performance is also poorer than that of YOLOv7. There are still gaps in research. The method of human action recognition using YOLOv7 are few. In response to this situation, we use YOLOv7 as the basis to achieve human action recognition, and the method we use is described in Chapter 3.

Chapter 3
Methodology

In this chapter, we introduce the deep learning methods used in this thesis. This chapter will focus on the details of deep learning methods and algorithms used in human action recognition. In addition, datasets suitable for this method will be presented.

3.1 YOLO v7

After YOLO was proposed, it was a typical representative of using the one-stage method for classification and recognition. YOLO-based methods are known for fast speed. However, while achieving high running speed, YOLO inevitably sacrifices part of the accuracy. In recent years, YOLO-based methods have also been rapidly disseminated due to various deep learning techniques being proposed. A variety of different YOLO methods improve the accuracy of YOLO while maintaining speed. YOLOv7 is the latest method based on YOLO. Figure 3.1 shows the simplified structure of YOLOv7.

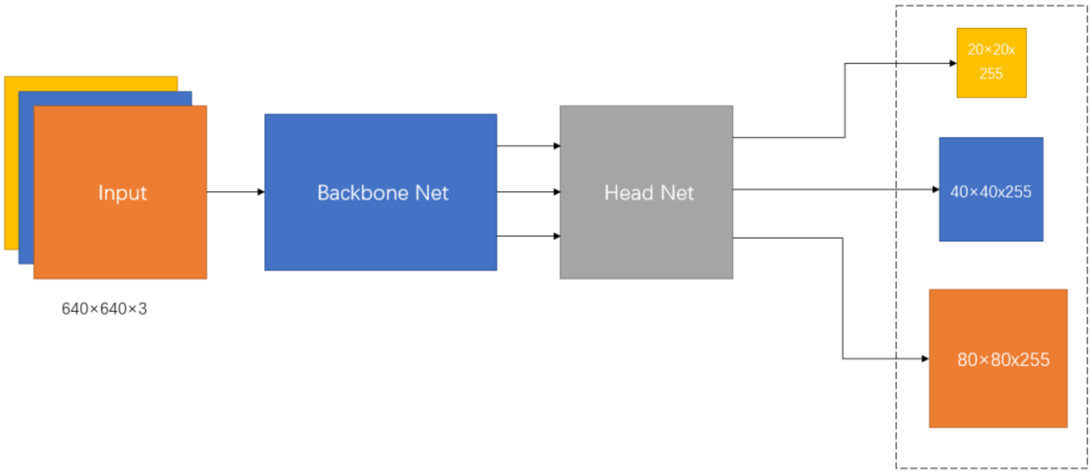


Figure 3.1: The simplified structure of YOLOv7 model

From the network architecture of YOLOv7, the network consists of three parts, which are input, backbone, and head. Among them, Backbone is responsible for extracting features, and head is responsible for predicting object categories and bounding boxes. Unlike YOLOv5, YOLOv7 combines the neck layer and head layer of the network, collectively referred to as the head layer. We will explain the specific structure of YOLOv7 in the next section (Wang, Bochknovskiy, & Liao, 2022).

The basic recognition process of the network can be summarized as: (1) Preprocess the input image and align it into an RGB image with a size of 640x640. (2) Input the picture into the backbone network, and generate different three-layer outputs through the backbone network. Then in the head layer, the three-layer output of the backbone network continues to be output as three layers of feature maps of different sizes. Finally, after the processing of RepVGG block and Conv, the output is predicted and the result is output.

3.1.1 Backbone

The specific structure of the YOLO v7 backbone layer is shown in Figure 3.2. Basically, it consists of the CBS module, the E-ELAN layer, and the MP layer.

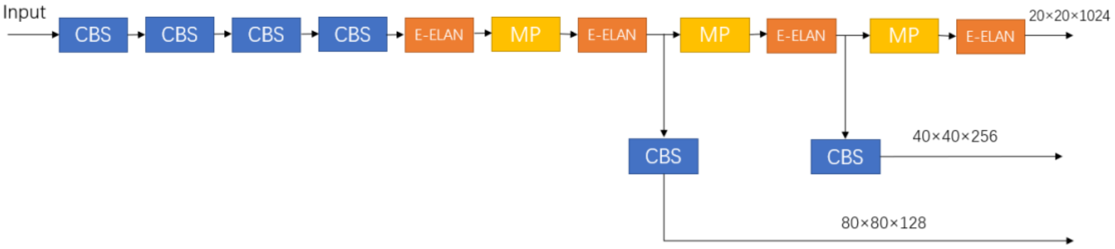


Figure 3.2: The structure of backbone

The CBS module is composed of a Conv layer, a Batch normalization (BN) layer, and a SiLU layer. where SiLU is an activation function. The full name is the sigmoid-weighted linear unit.

$$SiLU(x) = x \cdot Sigmoid(x) \tag{3.1}$$

$$Sigmoid(x) = \frac{1}{1+e^{-x}} \tag{3.2}$$

In Eq. (3.1), we see that SiLU activation function is obtained by multiplying the input by the Sigmoid function. Among them, the sigmoid function is also one of the common activation functions, and the function is expressed as shown in Eq. (3.2).

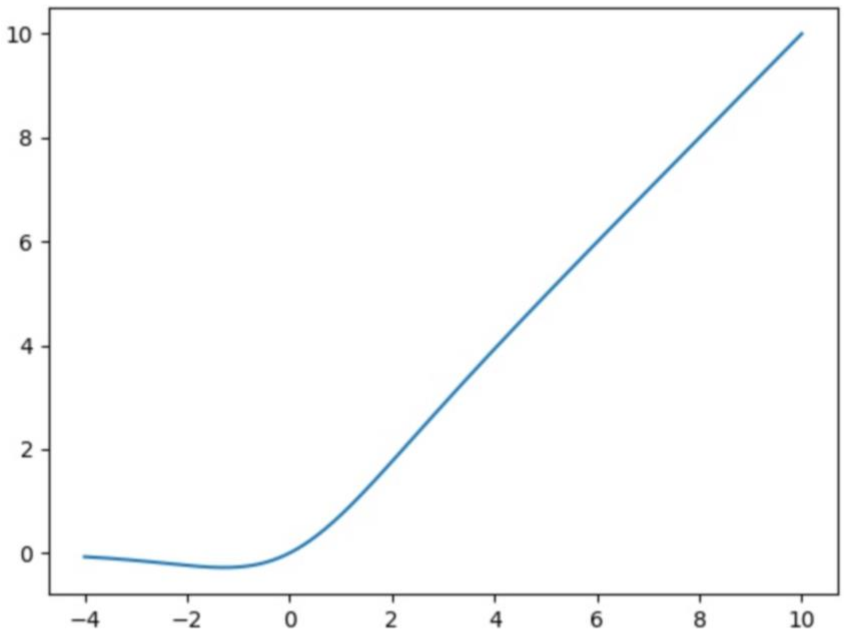


Figure 3.3: The curve of SiLu

By controlling the size and stride of the convolutional layer, CBS has different sizes. After four CBS modules, the feature map will become 160×160×128 size. Then input into the E-ELAN module.

The E-ELAN module is composed of multiple CBSs. The feature size does not change after passing through the E-ELLAN module. However, the learning ability of the network through E-ELAN will be improved, and more features can be learned. The structure of E-ELAN is shown in Figure 3.4.

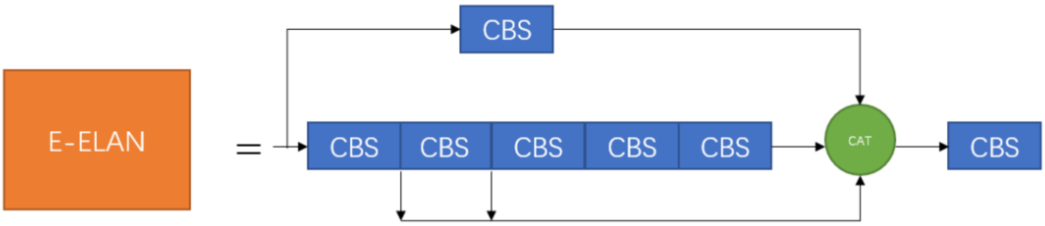


Figure 3.4: The structure of E-ELAN

From Figure 3.4, E-ELAN will generate four features, and E-ELAN will stack the obtained four features together to obtain the final feature extraction result. Then, this feature is input into the MP layer for downsampling. The structure diagram of MP is shown in Figure 3.5.

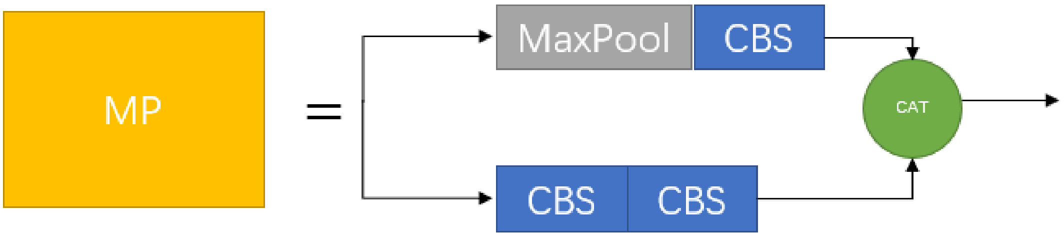


Figure 3.5: The structure of MP

The MP module consists of two branches together. The first branch goes through a maxpool operation, and then goes through a 1×1 convolution to get a sampling result. The second one goes through a 1×1 convolution, and then a 3×3 convolution with a stride of 2 to get another sampling result. Finally, MP will add the results of the two branches together to get the result.

Overall, after the input enters the backbone, it first enters 4 CBSs to obtain the feature map with the changed size. Then enter the E-ELAN structure to improve the network's learning ability of features while keeping the size unchanged. Then input into the structure of three MP + E-ELAN to get three feature maps of different sizes for the next head.

3.1.2 Head

The specific structure of Head is shown in Figure 3.6. It can be seen from the figure that Head is mainly composed of SPPCSPC structure, E-ELAN, MP structure and REP structure.

The SPPCSPC module is composed of two modules, SPP and CSP. As shown in Figure 3.7, the role of SPP is to enhances the receptive field so that the algorithm can adapt to images of different resolutions. It obtains various receptive fields through maximum pooling. The CSP module reduces the amount of computation. This is because the feature is first to split into two parts in this module. Then one is for conventional processing, and the other is for SPP structure processing. Finally, merge the two parts together. This method not only reduces the amount of calculation by half, making the speed faster but also improves the accuracy.

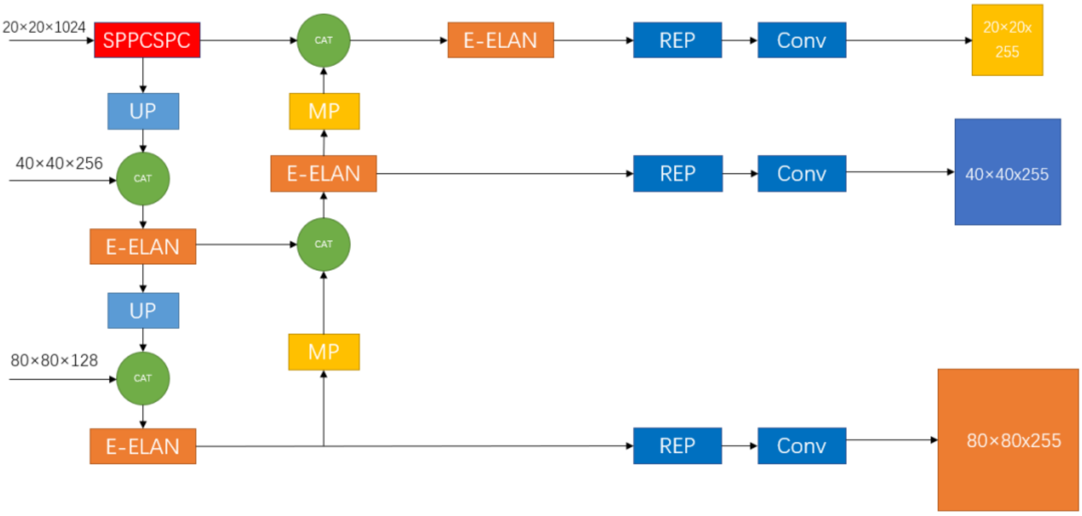


Figure 3.6: The structure of head

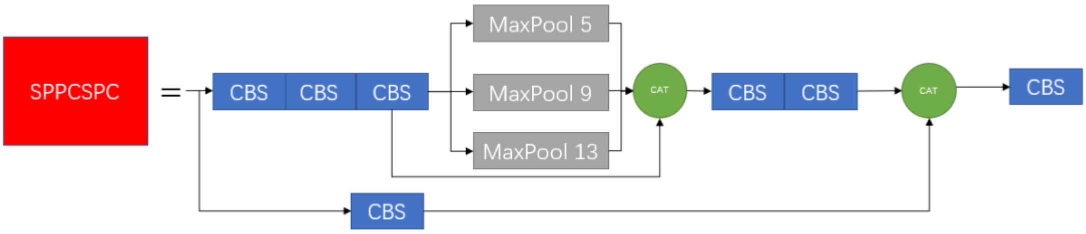


Figure 3.7: The structure of SPPCSPC

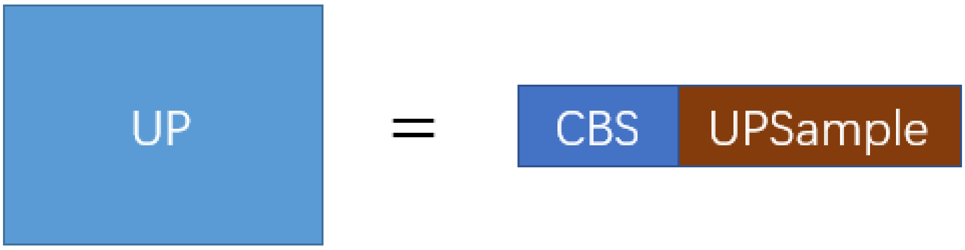


Figure 3.8: The structure of UP

The UP module consists of a CBS and a UPSample. The UPSample module is a module that uses nearest neighbor interpolation for upsampling. As shown in Fig. 3.8. The CAT module is responsible for stacking multiple features together to get the final feature result. The E-ELAN in head structure is analogous to that in the backbone. The difference is that the features output by E-ELAN in the head are composed of five features, not the four in the backbone. This is shown in Figure 3.9.

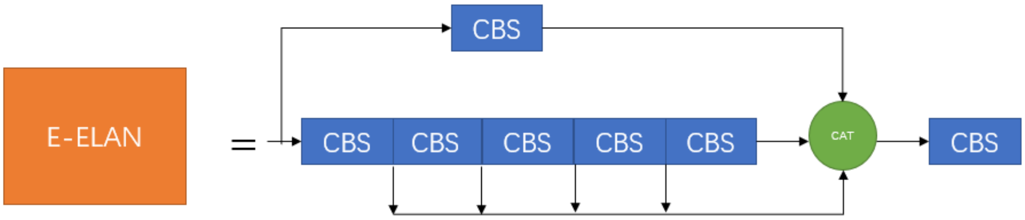


Figure 3.9: E-ELAN structure in head

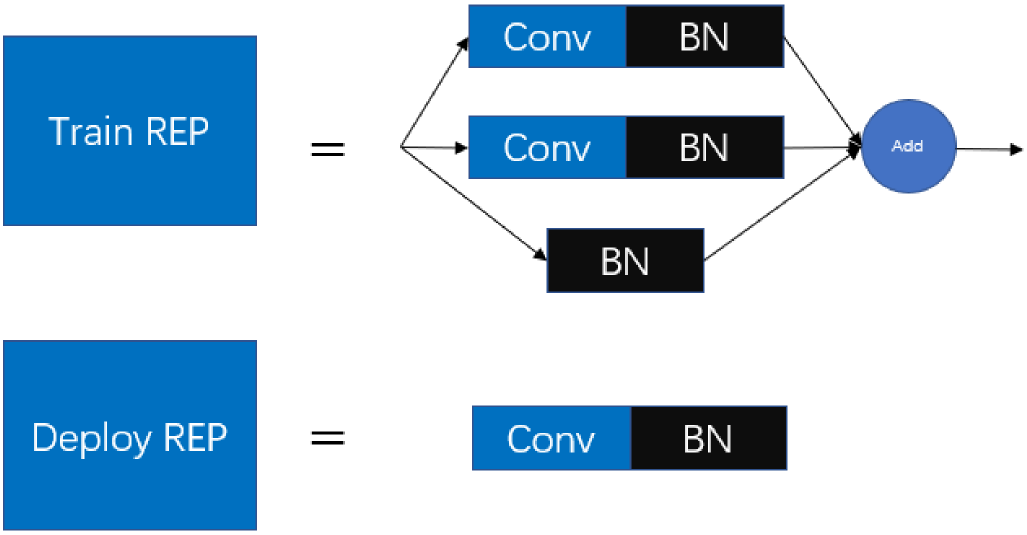


Figure 3.10: The structure of REP

The MP structure is just a change in the ratio of the number of channels. This causes the number of channels to change after the feature passes through the MP of the head, but does not change in the backbone.

There are two types of REP modules. One is the training module and the other is the deploy module. The training modules of the REP module are all composed of convolutions. There are three branches in the structure, the top branch is a 3 × 3 convolution for feature extraction. The middle branch is a 1×1 convolution for smoothing features. The last branch does not do convolution operations and directly input transmission. Finally, the results are added together. The deploy module of REP only contains a 3×3 convolution with a stride of 1. It is converted from re-parameterization of the training module as shown in Figure 3.10.

In the whole head, the three feature maps input by the backbone are output to the REP module after being processed by SPPCSPC, CBS, MP, and other structures. Output three different prediction results through three REP and conv layers.

3.2 Convolutional Block Attention Module (CBAM)

CBAM is an attention mechanism proposed (Woo et al. 2018). This attention mechanism combines spatial and channel attention mechanism modules. Compared with SENet which only focuses on the channel mechanism, CBAM achieves better results. The mechanism consists of two parts, the channel attention module, and the spatial attention module. As a generic module, CBAM can be loaded into any CNN. Because of this, CBAM has become a common attention mechanism. Its structure diagram is shown in Figure 3.11, which clearly indicates the structure of the attention module. Firstly, the importance of different channels, that is, the channel weights, is obtained. Then all feature maps are compressed into one feature map to obtain the weights of spatial features (Woo, Park, Lee, & Kweon, 2018).

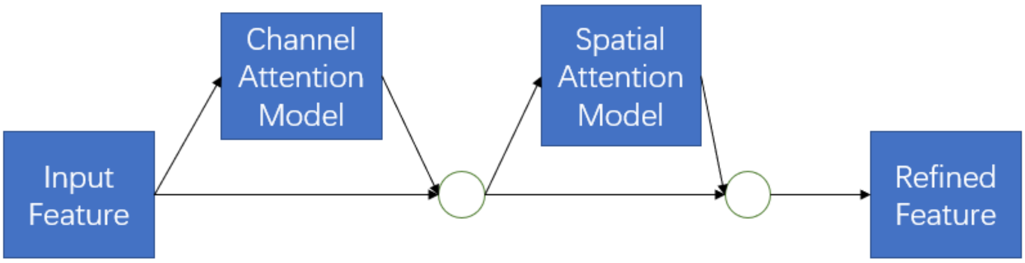


Figure 3.11: The simplified structure of CBAM

3.3 SimAM

SimAM attention mechanism is also an attention mechanism that pays attention to both channel attention and spatial attention modules. But this attention mechanism is different from other mechanisms in that this mechanism can derive 3D attention weights without the assistance of additional parameters. In the attention mechanism at that time, the generation of attention weights often requires the assistance of additional sub-networks. It is also feasible to infer 3D weights directly from current neurons. Therefore, they defined an energy function to help them efficiently infer such three-dimensional weights. A single SimAM module can be a computational unit that can be used to enhance the expressiveness of features in convolutional neural networks. It can take any intermediate feature as input and convert it into a feature with constant size and more expressive power. A simple structure is shown in Figure 3.12. By comparing with other attention mechanisms, the final experimental results show that the attention mechanism is very effective (Yang, Zhang, Li, & Xie, 2021).

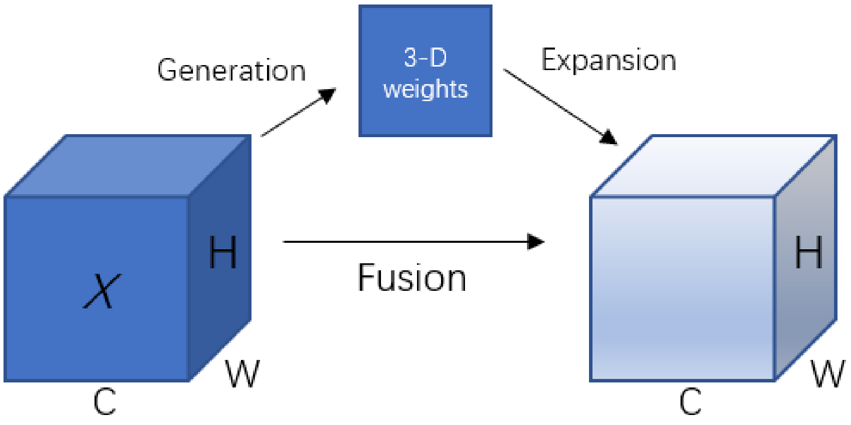


Figure 3.12: The simplified structure of SimAM

3.4 Research Designing

Human action recognition is the core of this thesis. In this thesis, we take use of YOLOv7 as the base network for human action recognition. And use the attention mechanism as the complementary structure of the network. The two jointly build a model of YOLOv7+ attention mechanism as a typical method for human action recognition in this thesis. In terms of overall design, this thesis has designed three disparate models. They are YOLOv7+CBAM, YOLOv7+SimAM and YOLOv7+CBAM+SimAM respectively. Among them, YOLOv7+CBAM and YOLOv7+SimAM are implemented by adding the attention mechanism to YOLOv7 respectively. YOLOv7+CBAM+SimAM is implemented by adding these two methods to YOLOv7. By using these three models, we will realize the recognition of the five actions of clutching, boxing, walking, waving, and running.

In addition, since the underlying network structure is YOLO. Therefore, in this thesis, we create a new dataset YOLOv7 Action based on KTH (Schuldt, Laptev, & Caputo, 2004), UCF-101 (Soomro, Zamir, & Shah, 2012), Weizmann (Gorelick, Blank, Shechtman, Irani, & Basri, 2005), HDBM (Kuehne, Jhuang, Garrote, Poggio, & Serre, 2011), UTKinect-Action3D (L Xia, 2012), MSR Action datasets (Yuan, Liu, & Wu, 2011).

Among them, there are 6 actions in the KTH dataset are performed by 25 people multiple times in indoor and outdoor environments. The data in the UCF-101 dataset is real action video data collected from YouTube. There are 101 action categories in the dataset, and the recorded actions tend to have a cluttered background. The Weizmann dataset consists of 90 action video sequences in which 9 different people perform 10 different natural actions in the dataset. The HDBM dataset has 51 action categories, mostly collected from movies. UTKinect-Action3D has three different types of action data: RGB, depth and bone joint position, and there are ten action categories in total. In this thesis, we only use RGB data as a complement to other datasets. The dataset incorporates appropriate action videos from these six datasets, then converts these video data into data that can be trained in YOLOv7 as the training set for this thesis. Figure 3.13 shows an example image of the dataset used in this thesis.

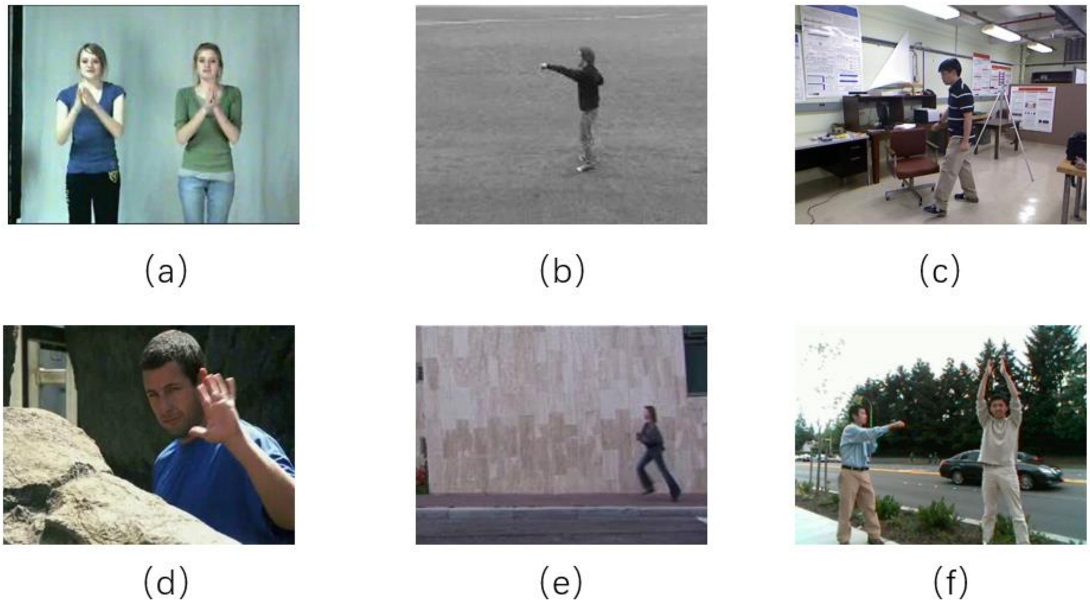


Figure 3.13: video frames as examples from this dataset. (a) Clapping, (b) Boxing, (c) Walking, (d) Waving, (e) Running, (f) Boxing and Waving

3.4.1 YOLOv7 + CBAM

Among the recognition models, YOLOv7 is the core part of the whole network. All calculations will go through YOLO's backbone and head. The introduction of the attention mechanism is to modify the backbone and head of YOLOv7. Since YOLO is a network structure based on a convolutional neural network. It is very common to introduce attention mechanism to enhance the performance of the network.

In the model of YOLOv7+CBAM, CBAM is added in both backbone and head. In the backbone, we add CBAM between the CBS module and the E-ELAN module. In addition, CBAM was also added after the other E-ELAN of the backbone. In the header, CBAM is added to the E-ELAN. We named this structure E-ELANCBAM. Then use this new module to replace the E-ELAN module in the CAT module and the UP module. The specific improved structure is shown in Figure 3.14. In E-ELANCBAM, we are use of CBAM to replace the last CBS in the original E-ELAN. 3.15 shows the specific structure.

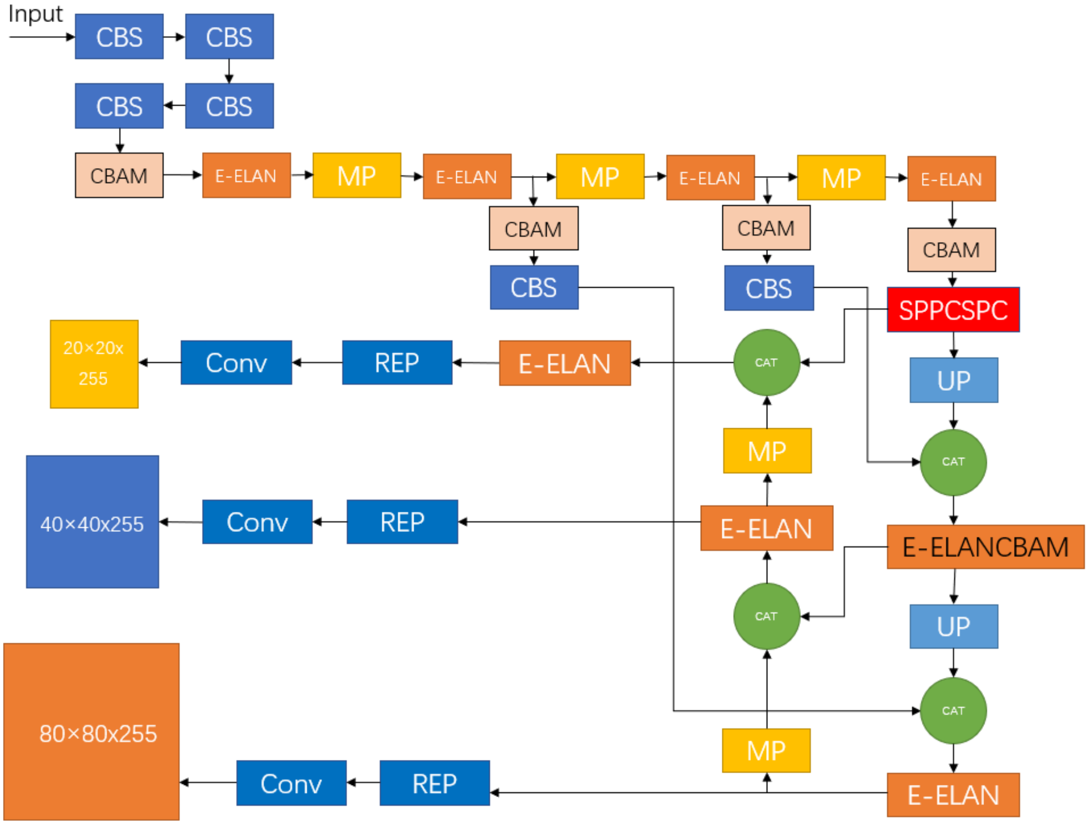


Figure 3.14: The structure of YOLOv7+CBAM

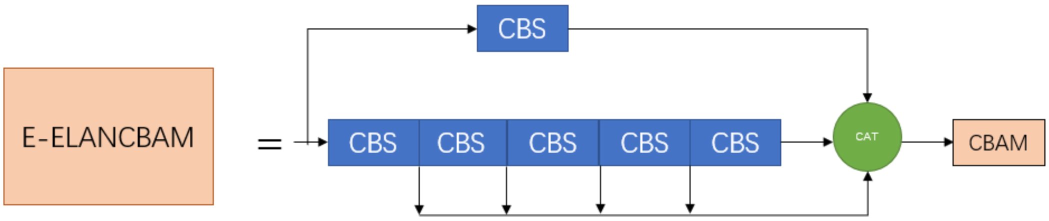


Figure 3.15: The structure of E-ELANCBAM

3.4.2 YOLOv7 + SimAM

Like the YOLOv7+CBAM model, the SimAM module is still placed in the backbone and head in the structure of the YOLOv7+SimAM model. In terms of specific improvements, we firstly add a SimAM module after the four CBS modules and before the first E-ELAN module. Then, we add the SimAM module after the remaining three E-ELANs in the backbone. This is an improvement on the backbone of the YOLOv7+SimAM model. In the head, we add the SimAM module to the E-ELAN module between the CAT module and the UP module. We name this model E-ELANSimAM. The overall structure is shown in Figure 3.16. In E-ELANSimAM, we are use of SimAM model to replace the last CBS in the original E-ELAN. 3.17 shows the specific structure.

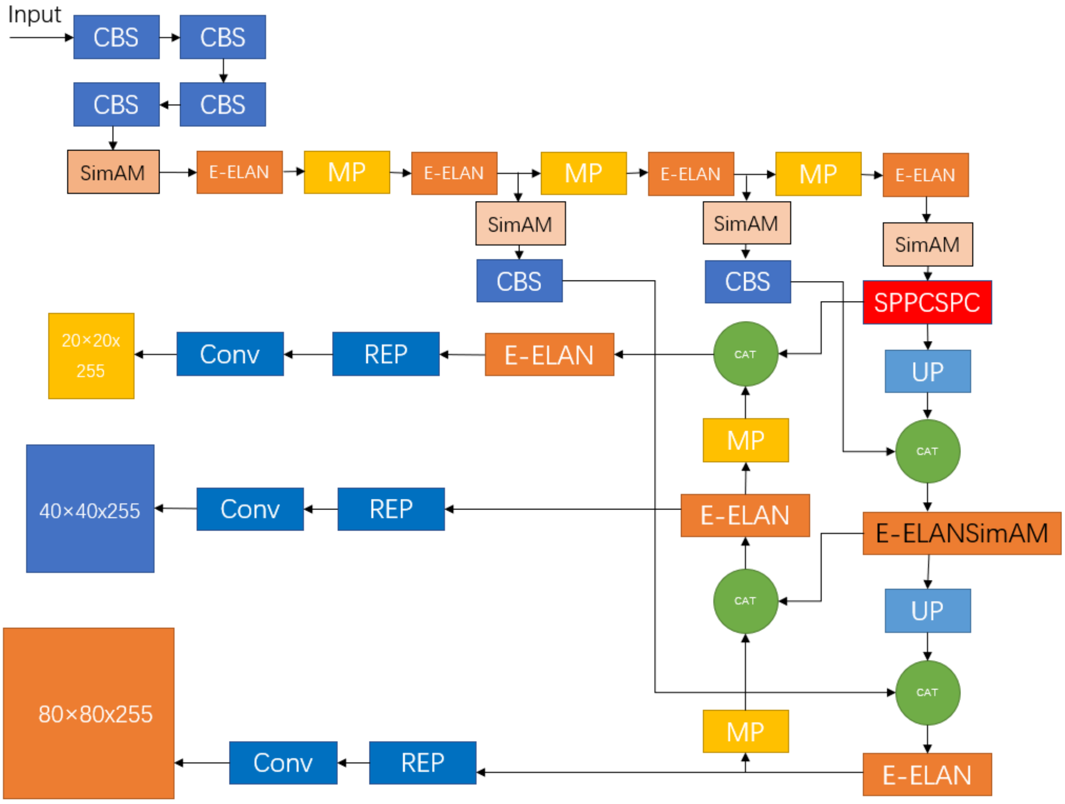


Figure 3.16: The structure of YOLOv7+CBAM

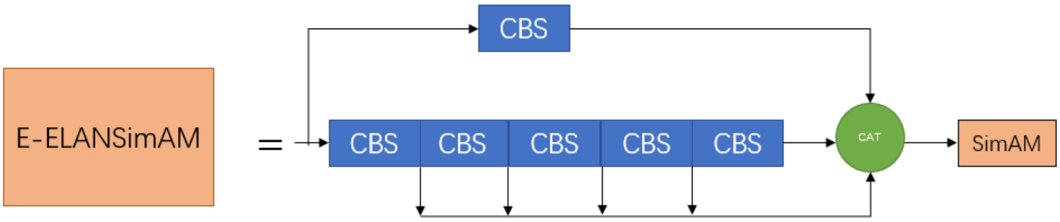


Figure 3.17: The structure of E-ELANSimAM

3.4.3 YOLOv7 + CBAM+SimAM

The YOLOv7 + CBAM + SimAM model is based on a combination of the first two models. The improvement of this model is based on the improvement of the previous two models. The purpose is to apply the first two attention mechanisms together in the YOLOv7 network. We add E-ELANCBAM to the backbone of the YOLOv7 + CBAM +

SimAM model. We replaced the original E-ELAN module of the backbone with this module. In the head, the E-ELAN SimAM module is introduced. Still this module was added to replace the E-ELAN module between the CAT module and the UP module. The overall structure of YOLOv7+CBAM+SimAM is shown in Figure 3.18.

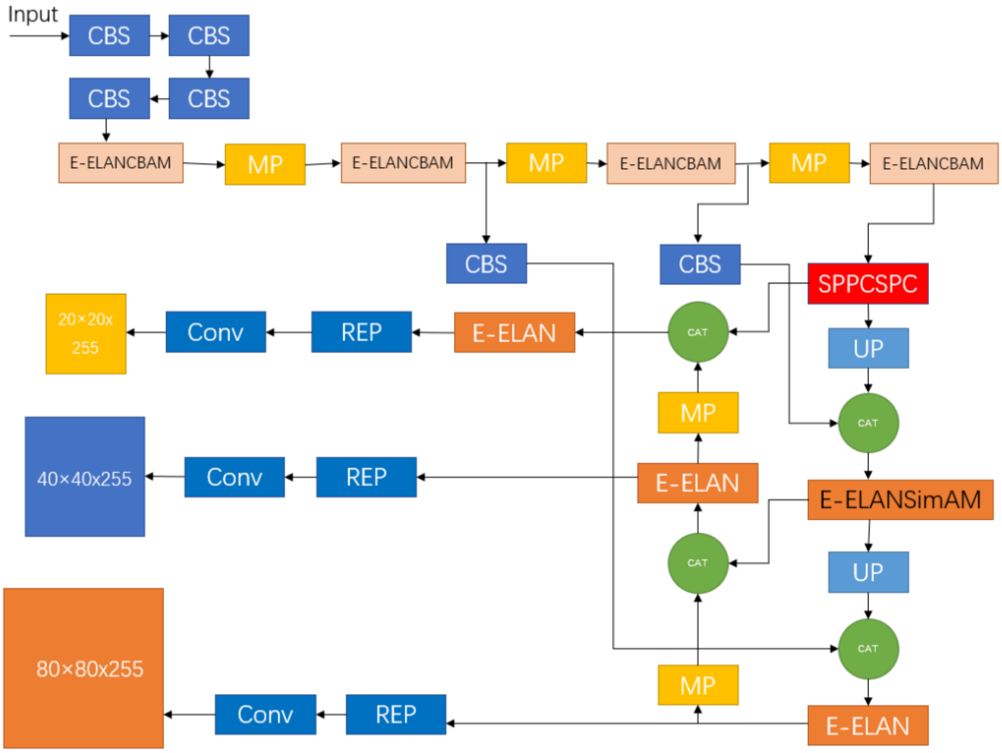


Figure 3.18: The structure of YOLOv7+CBAM+SimAM

3.4.4 Algorithms

Whether the network performance is excellent or not is often closely related to the training process. In this thesis, the network structure is based on YOLOv7. Therefore, the loss function of YOLOv7 itself is chosen to be used in the use of the loss function. There are three types of loss functions in YOLOv7 bounding box loss, confidence loss and classification loss. These three losses can be defined as,

$$L_{all} = L_{box} + L_{obj} + L_{cls} \tag{3.3}$$

Among them, L_{box} is responsible for viewing the error between the predicted frame and the calibration frame, that is, the bounding box loss. L_{obj} is responsible for calculating the confidence loss of the network, which is called object confidence loss. L_{cls} is responsible for calculating whether the anchor box and the corresponding calibration classification are correct, that is, the classification loss.

Classification Loss and Object Confidence Loss

Both the confidence loss and class loss functions in YOLOv7 are based on the Binary Cross Entropy loss (BCE) function with sigmoid. The BCE loss function is shown as eq. (3.4).

$$BCE = -\sum_{i=1}^{N} y^{(i)} \log \hat{y}^{(i)} + \left(1 - y^{(i)}\right) \log \left(1 - \hat{y}^{(i)}\right) \tag{3.4}$$

where $\hat{y}$ represents the probability that the model predicts that the *i* th sample is a certain class, and $y^{(i)}$ represents the label. Therefore, there will be different calculation methods for label ‘0’ or label ‘1’. For multiclassification problems, it is equivalent to expanding the dimension of the label itself.

The object confidence loss is calculated using pairs of samples obtained by matching positive samples. The object confidence loss function is shown in Eq. (3.5).

$$L_{obj}(p_o, p_{iou}) = BCE_{cls}^{sig}\left(p_o, p_{iou}; w_{obj}\right) \tag{3.5}$$

where p_o is the object confidence score in the prediction box, p_{iou} is the iou value of the predicted box and its corresponding target box. In the function, p_{iou} is used as ground-truth and po to calculate the final object confidence loss through BCE. The classifcation loss is analogous to the confidence loss. The classification loss function is shown as Eq.(3.6),

$$L_{cls}\left(c_p, c_{gt}\right) = BCE_{cls}^{sig}\left(c_p, c_{gt}; w^{cls}\right) \tag{3.6}$$

where c_p is the class score of the predicted box, c_{gt} is the one-hot representation of the target box category. In the function, the two are calculated by BCE to obtain the final class loss.

Bounding Box Loss

The loss function of bounding box loss is CIoU Loss. This loss function is proposed based on IoU (Yu, Jiang, Wang, Cao, & Huang, 2016). The IoU is shown in Eq. (3.7),

$$L_{IoU} = -\ln(iou) \tag{3.7}$$

The bounding box loss is shown as Eq. (3.8),

$$CIOU_{LOSS} = 1 - CIOU - (IOU - \frac{d_o^2}{d_c^2} - \frac{v^2}{1-IOU+v}) \tag{3.8}$$

where d_o is the distance between the center point of the target frame and the center point o of the prediction frame, d_c is the diagonal distance of the target frame. v is a parameter to measure the consistency of the aspect ratio. The definition of v is shown in eq.(3.9).

$$v = \frac{4}{\pi}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w^{p}}{h^{p}})^2 \tag{3.9}$$

where w^{gt} and h^{gt} are the width and height of the real target box. w^p and h^p are the width and height of the prediction box.

3.5 Evaluation Methods

After the recognition of human actions is achieved through the model, the final performance of the results needs to be evaluated. We will perform ablation experiments on the obtained results to determine the performance of the model. There are many classification methods that can be used in this project. Such as accuracy, recall, precision, etc. In order to evaluate the model performance, the various evaluation methods used in this project will be laconically introduced next.

Before introducing evaluation methods, we need introduce four different concepts. In order to describe the method used more simply, this thesis calls both the predicted result and the real result true as TA. The prediction result is true, and the real result is false is called FA. The prediction result and the real result are both false and called FA. The prediction result is false and the real result is true is called FB. Most of the evaluation indicators are calculated by the four.

The first evaluation metric is accuracy. Accuracy is the percentage of correct predictions in all quantities. It is shown in eq.(3.10).

$$Accuracy = \frac{(TA + TB)}{(TA + TB + FA + FB)} \tag{3.10}$$

The second evaluation metric is Precision. Precision is the proportion of correctly predicted results out of all predicted true results. It is shown in eq.(3.11).

$$Precision \ = \frac{TA}{(TA + FA)} \tag{3.11}$$

The third evaluation metric is Recall. Recall is the number of true predictions checked out of all true results. It is shown in eq.(3.12).

$$Recall \ = \frac{TA}{(TA+FB)} \tag{3.12}$$

F1 is the fourth evaluation metric. It is calculated by precision and recall. F1 is often used as a complement to both, the bigger the better. It is shown in eq.(3.13).

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \tag{3.13}$$

The fifth metric is the confusion matrix. The confusion matrix displays TA, TB, FA, and FA in the form of graphs, making the results more intuitive. As shown in Table 3.1.

Table 3.1: Confusion matrix

True Situation	**Prediction Situation**	
	Positive	Negative
Positive	TA	TB
Negative	FA	FB

The sixth indicator is *mAP*, through which the detection ability of the trained model on all categories can be determined. mAP is calculated by calculating the average value of all AP_s. In a broad sense, AP is the area under the PR curve obtained with recall as the abscissa and precision as the ordinate. It is shown in eq.(3.14).

$$mAP = \frac{\sum_{i=1}^{K} AP_i}{K} \tag{3.14}$$

Image bounded by Precision and Confidence. Image bounded by Recall and Confidence. Image bounded by F1 and confidence. These images can all be used to evaluate the performance of the model. In addition, the runtime of the recognition action can also be used as one of the evaluation criteria. The YOLO algorithm has always been known to run fast. Running time can be regarded as an evaluation index of efficiency. In general, a model with a shorter runtime and higher recognition accuracy is better. All these methods can be used as evaluation indicators.

Chapter 4
Results

The main content of this chapter is the experimental results and comparison of human action recognition. In addition, the environment for the experiment will be briefly described. At the end of this chapter, the limitations and limitations of the project are discussed.

4.1 Data Collection and Experimental Environment

The main purpose of this thesis is to obtain a model that can efficiently recognize human actions in video sequences. Thus, a suitable dataset is essential. There are several 50 videos in the dataset. Amongst them, 45 videos are 5 to 10 seconds in size, and each video data contains only one action. In the remaining video data, each data has multiple actions or more than one person performing actions. There are a total of 5 action classes in the dataset, such as clutching, boxing, walking, waving, and running. The dataset has a total of 1030 labels, the specific number of labels for each action is shown in Table 4.1.

Table 4.1: The number of labels

Action	**Labels**
Clapping	187
Boxing	213
Walking	252
Waving	188
Running	190

We have not yet created a test dataset for the model. In the test dataset, there are 130 images and 15 videos. The data in the dataset also comes from the 6 datasets mentioned in the previous chapter. These datasets were selected to contain data for 5 actions that could be identified. In order to ensure the accuracy of the test results, the data selection in the test set avoids the data used in the training set.

Most of the experiments for human action recognition projects are performed in Colab, such as training models. Testing is performed locally. The experimental training process is run on an A100-SXM4-40GB high-performance graphics card in a virtual environment. This drastically reduces the time it takes to train.

4.1.1 Human Action Recognition Train Result

In Chapter 3, we propose three different YOLO models. To facilitate comparison of results and determine whether the performance of the model has improved, we trained four models of YOLOv7, YOLOv7+SimAM, YOLOv7+CBAM, YOLOv7+CBAM+SimAMsige, respectively. Each model goes through 150 iterations during training, and the batch-size is set to 16. Finally got 4 different results.

The basic YOLOv7 network takes 1.5 hours for the network to complete 150 epochs of training. Figure 4.1 shows the trend of precision and recall. The abscissa in the figure is the number of training epochs. Recall fluctuates wildly in the middle. In the 150 epochs of YOLOv7 training, both tend to stabilize as the number of epochs enhances.

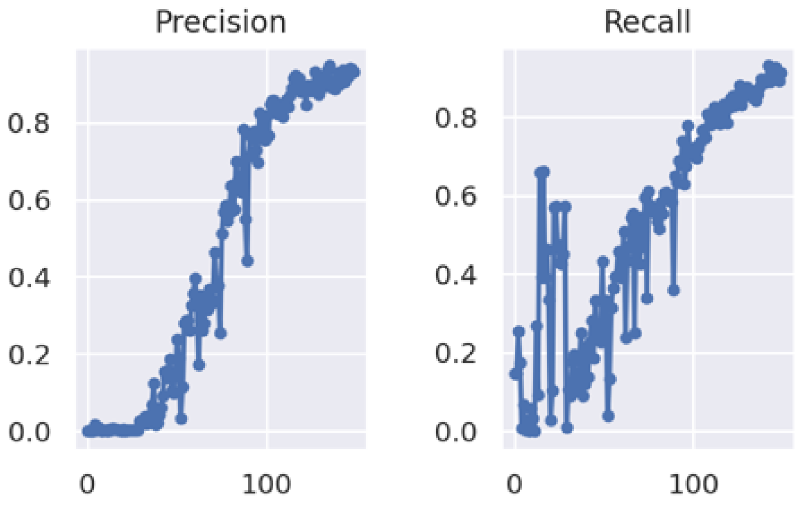


Figure 4.1: Precision and recall of YOLOv7

Figure 4.2 shows the changes in mAP values in the YOLOv7 model. The number after @ represents the threshold for judging iou as a positive or negative sample. mAP@.5: Indicates the average mAP with a threshold greater than 0.5. mAP@0.5:0.95 means from 0.5 to 0.95 in steps of 0.05, taking the average mAP over 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95. The value of mAP enhances gradually with the increase of epochs.

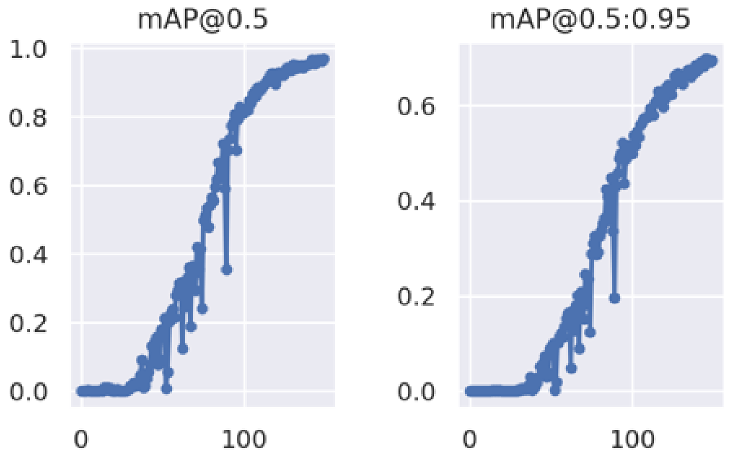


Figure 4.2: The mAP of YOLOv7

Figure 4.3 reflects the relationship between accuracy F1 and confidence of YOLOv7. In the machine learning problems of multi-classification, F1-score is often employed as the final evaluation method. The F1 and confidence results of the boxing curve are evidently worse than the overall findings in the figure. The results of all other action curves are superior to the average. While YOLOv7 achieves better F1 results between the confidence levels of 0.2 to 0.4. The highest F1 of YOLOv7 is 0.92.

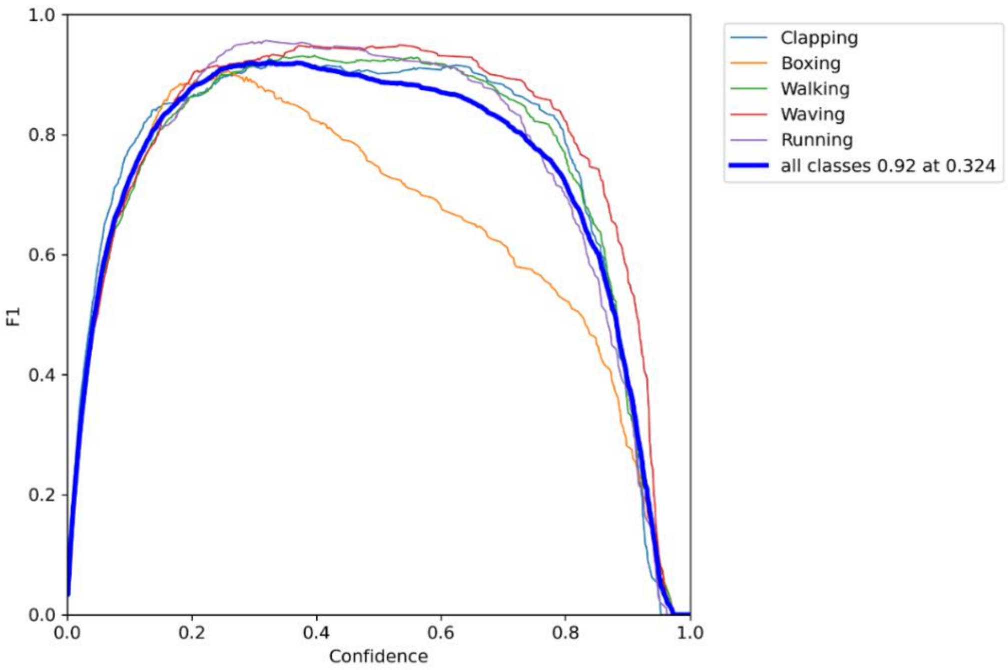


Figure 4.3: F1_curve of YOLOv7

Figure 4.4 shows the confusion matrix of YOLOv7. From the data on the confusion matrix, we get the overall accuracy of the YOLOv7 model. The accuracy of a single action can also be obtained directly. The specific results are that the accuracy of action “Clapping” is 0.95, the accuracy of action Boxing is 0.90, the accuracy of action “Walking” is 0.92, the accuracy of action “Waving” is 0.94, and the accuracy of action “Running” is 0.94.

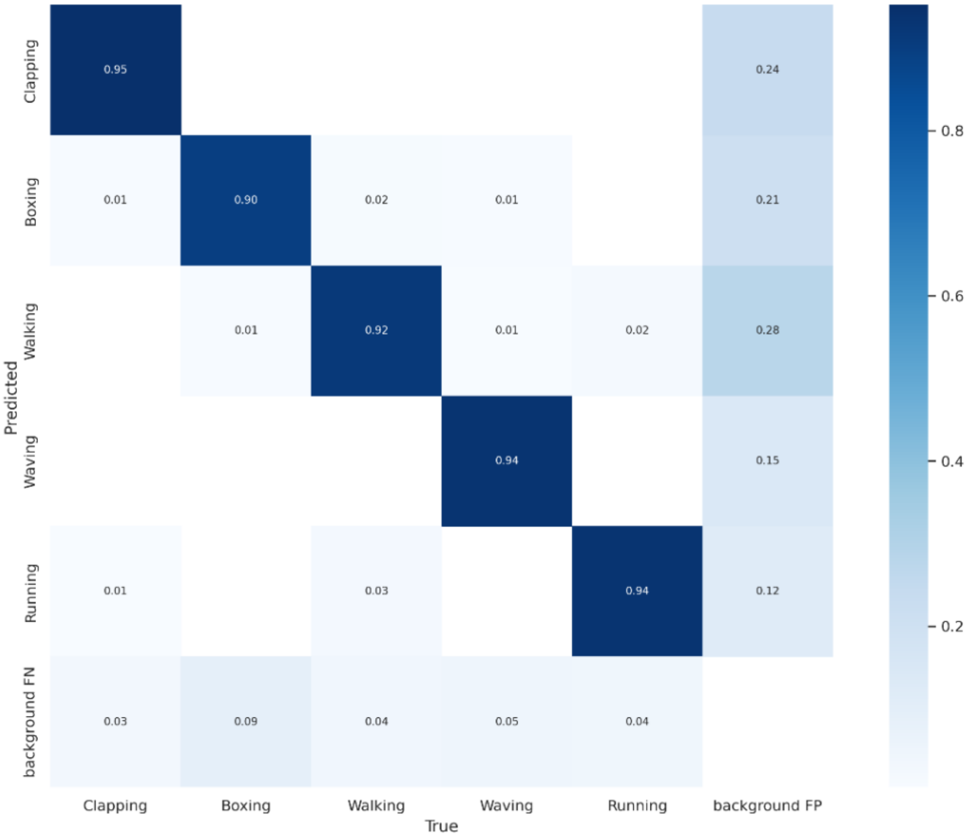


Figure 4.4: The confusion matrix of YOLOv7

Figure 4.5 shows the PR diagram of YOLOv7 which represents the relationship between precision and recall. The area enclosed by the curve is the mAP@0.5 value for that action. The mAP@0.5 of action “Clapping” is 0.98. The mAP@0.5 of action “Boxing” is 0.964. The mAP@0.5 of action “Walking” is 0.97. The mAP@0.5 of action “Waving” is 0.973. The mAP@0.5 of action “Running” is 0.963.

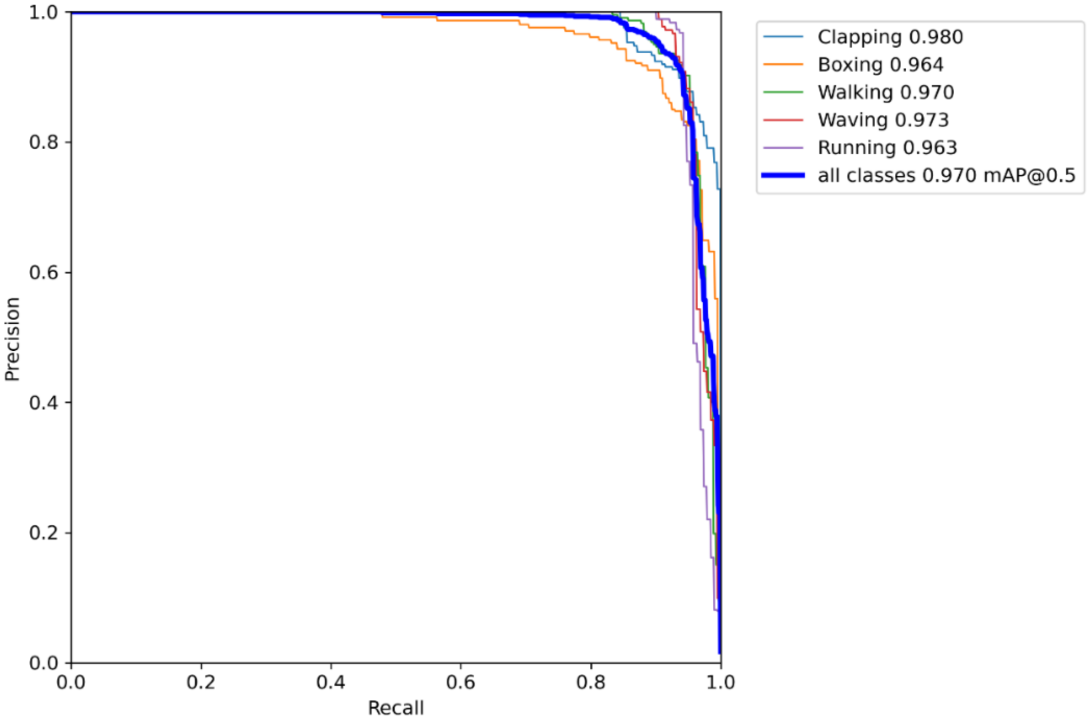


Figure 4.5: The PR_curve of YOLOv7

For the YOLOv7+CABM network, it takes 1.48 hours to complete the training of 150 epochs. Figure 4.6 shows the trend of precision and recall. In the 150 epochs of YOLOv7+CBAM training, both tend to stabilize as the number of epochs enhances. Figure 4.7 shows the changes in mAP values in the YOLOv7+CBAM model.

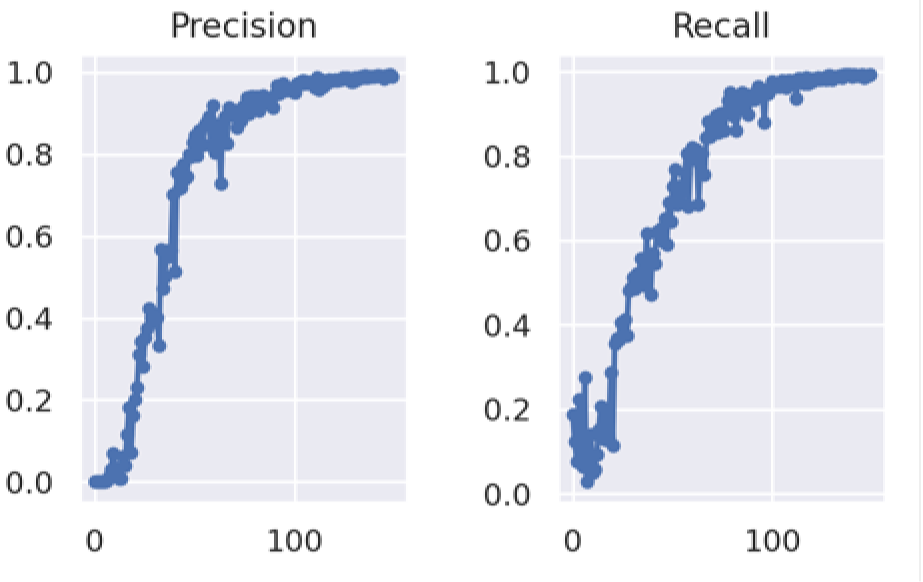


Figure 4.6: The precision and recall of YOLOv7+CBAM

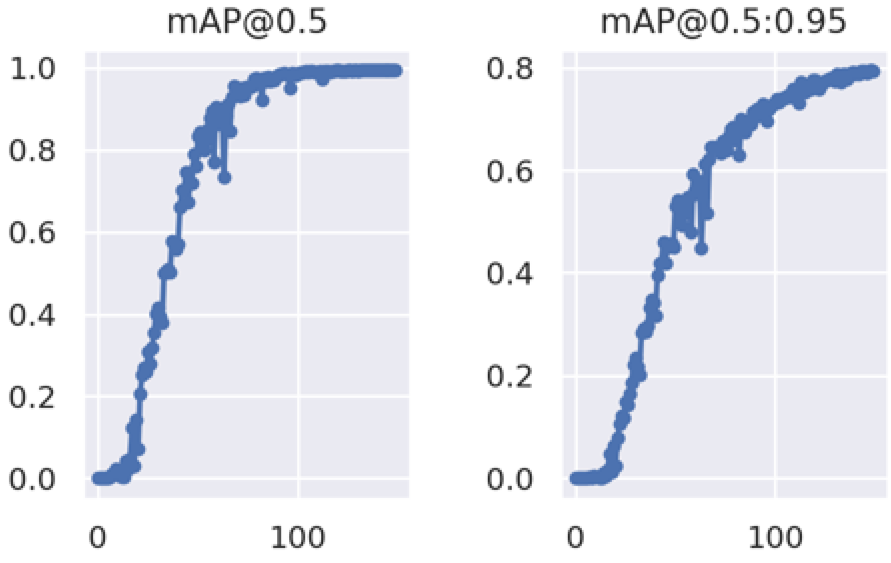


Figure 4.7: The mAP of YOLOv7+CBAM

Figure 4.9 indicates the relationship between accuracy F1 and confidence of YOLOv7+CBAM. The boxing curve of F1 and confidence results also are obviously worse than the figure's overall findings. All other action curves produce results that were in most cases better than average. YOLOv7+CBAM achieves better F1 results within the confidence interval of 0.4 to 0.6. The highest F1 of YOLOv7+CBAM is 0.99.

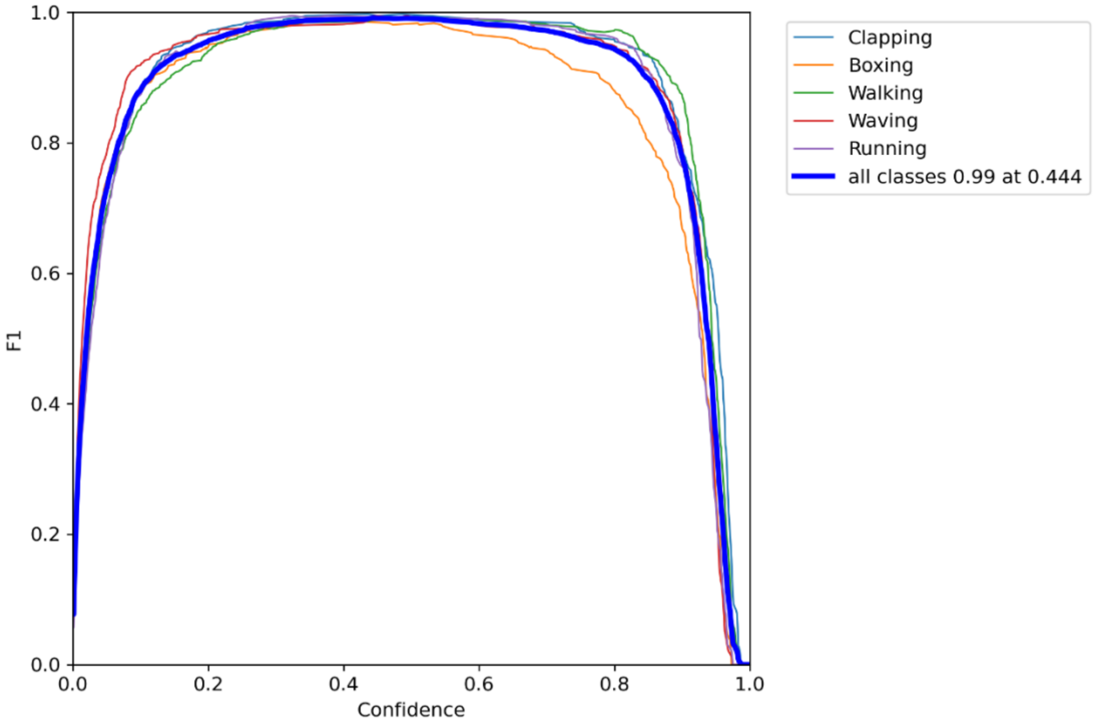


Figure 4.8: F1_curve of YOLOv7+CBAM

Figure 4.9 shows the confusion matrix of YOLOv7+CBAM. The specific results are that the accuracy of action “Clapping”, “Boxing”, and “Running” is 0.99, the accuracy of action “Walking” is 0.98, and the accuracy of action “Waving” is 1.00.

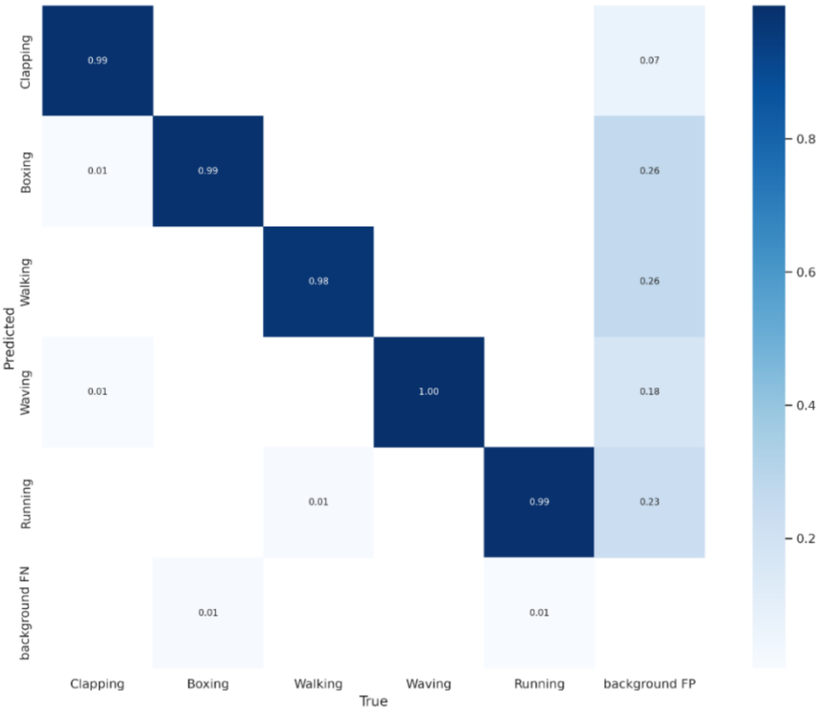


Figure 4.9 The confusion matrix of YOLOv7+CBAM

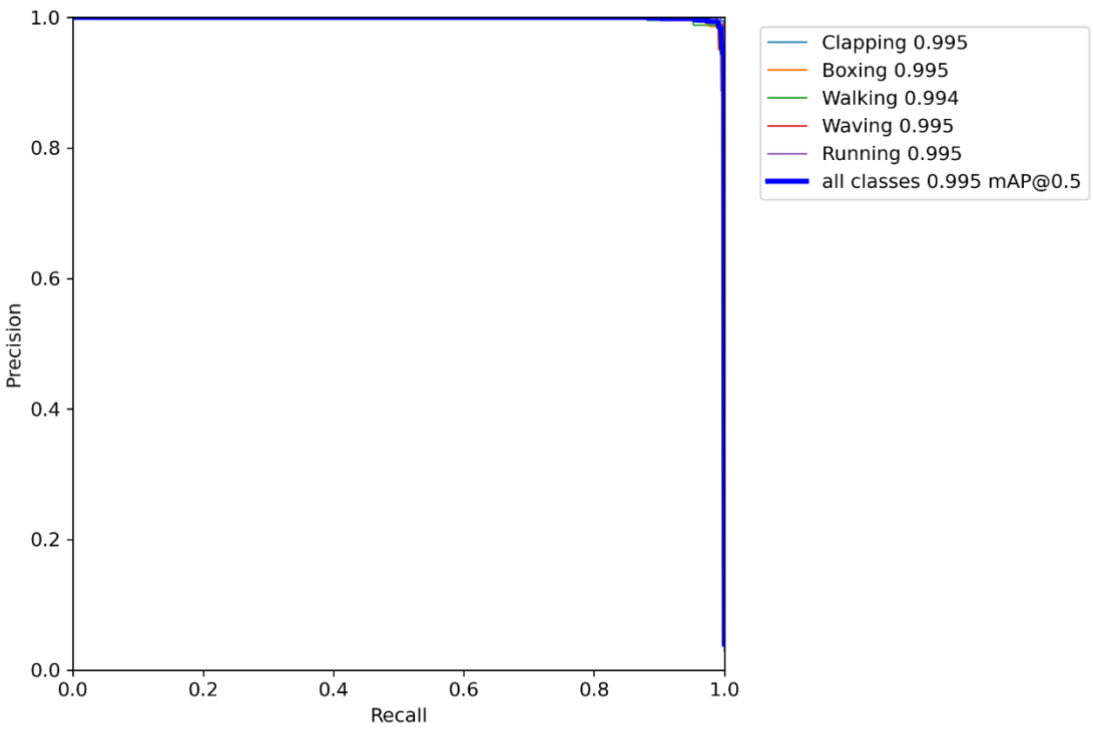


Figure 4.10 PR_curve of YOLOv7+CBAM

Figure 4.10 shows the PR diagram of YOLOv7+CBAM which represents the relationship between precision and recall. The area enclosed by the curve is the mAP value for that action. The mAP@0.5 of actions “Clapping”, “Boxing”, “Waving”, and “Running” is 0.995. The mAP@0.5 of action “Walking” is 0.994.

For the YOLOv7+SimAM network, it costs 1.43 hours to complete the training of 150 epochs. Figure 4.11 shows the trend of precision and recall. The precision has a high value at the beginning, and then gradually returns to normal. In the 150 epochs of YOLOv7 training, both tend to stabilize as the number of epochs enhances. Figure 4.12 shows the changes in mAP values in the YOLOv7+SimAM model.

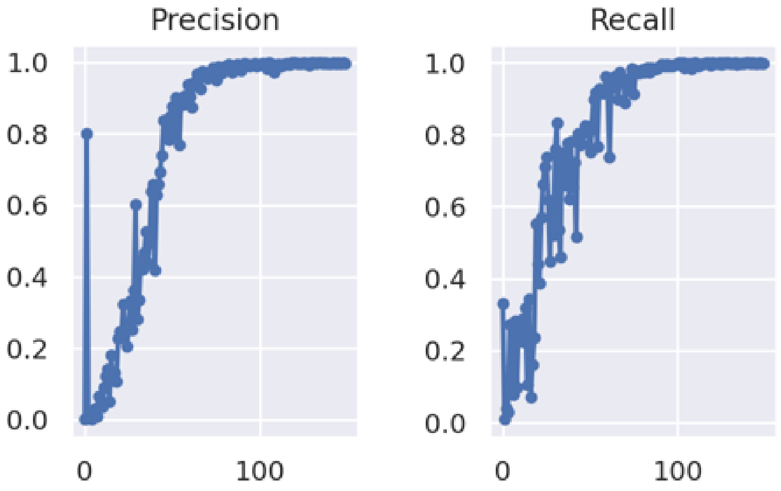


Figure 4.11 The precision and recall of YOLOv7+SimAM

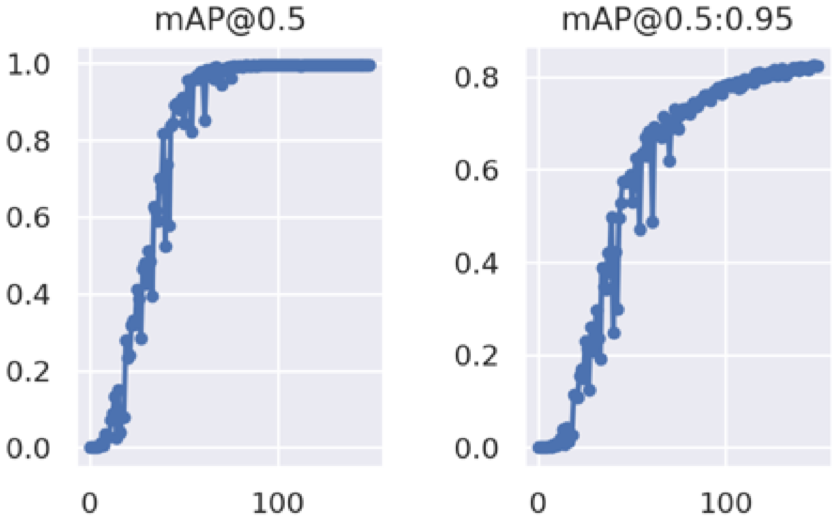


Figure 4.12 The mAP of YOLOv7+SimAM

Figure 4.13 shows the relationship between accuracy F1 and confidence of

YOLOv7+SimAM. YOLOv7+SimAM achieves a good F1 with a confidence interval of 0.4 to 0.8. The highest F1 of YOLOv7+SimAM is 1. Among them, the curve of action Boxing is slightly worse than other curves.

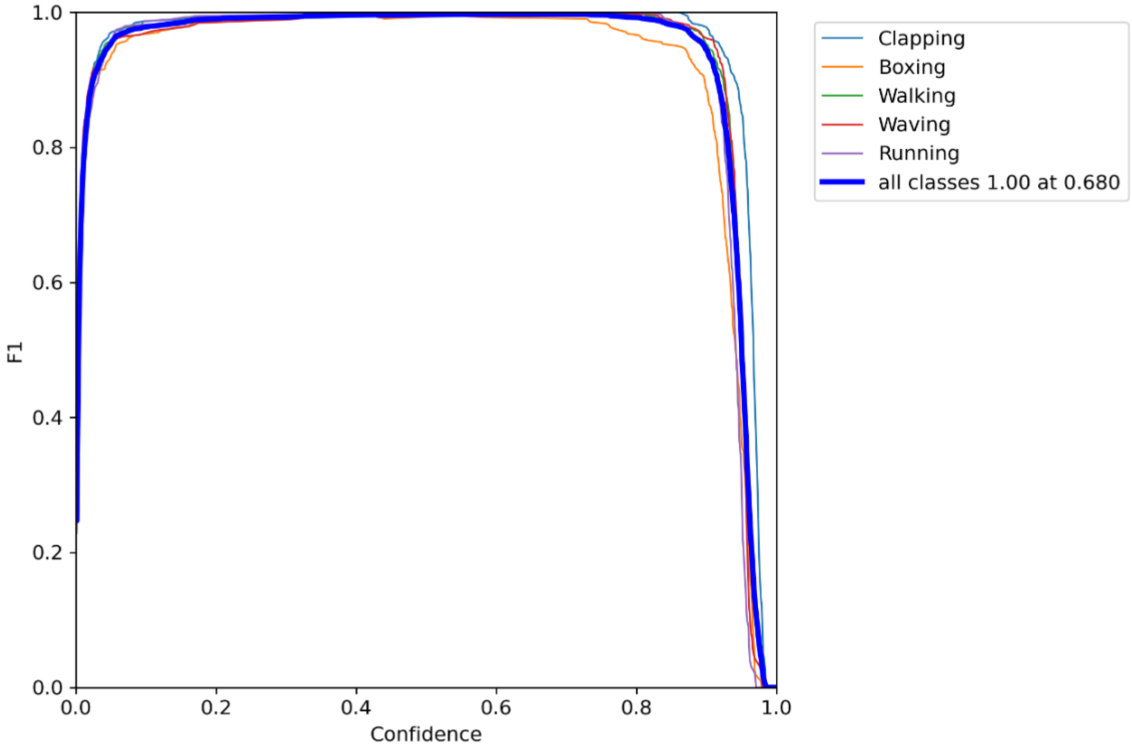


Figure 4.13: F1_curve of YOLOv7+SimAM

Figure 4.14 shows the confusion matrix of YOLOv7+SimAM. We get the accuracy of each action from the matrix. The specific results are that the accuracy of action "Walking" is 0.99, the accuracy of action "Boxing" is 0.98, and the accuracy of actions "Clapping", "Waving", and "Running" is 1.00.

Figure 4.15 shows the PR diagram of YOLOv7+SimAM, which represents the relationship between precision and recall. The mAP@0.5 of five actions are 0.996. In addition, we only see one overall curve from the graph, and all other curves are covered by the overall curve.

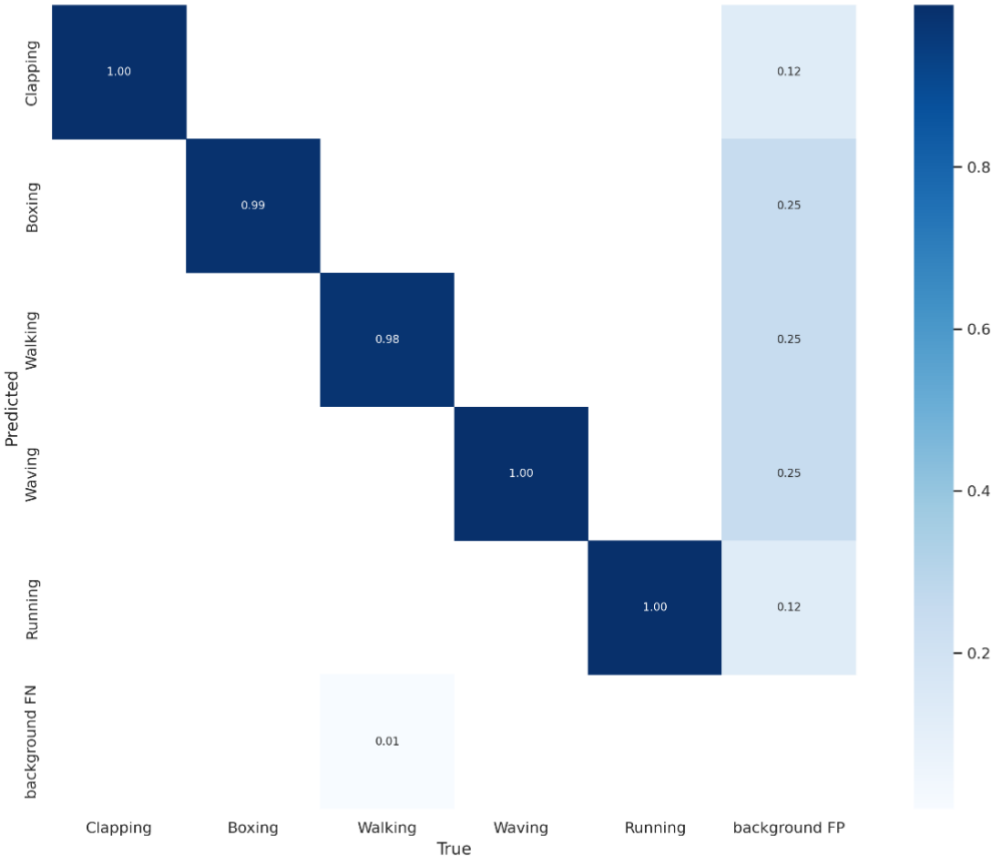


Figure 4.14: The confusion matrix of YOLOv7+SimAM

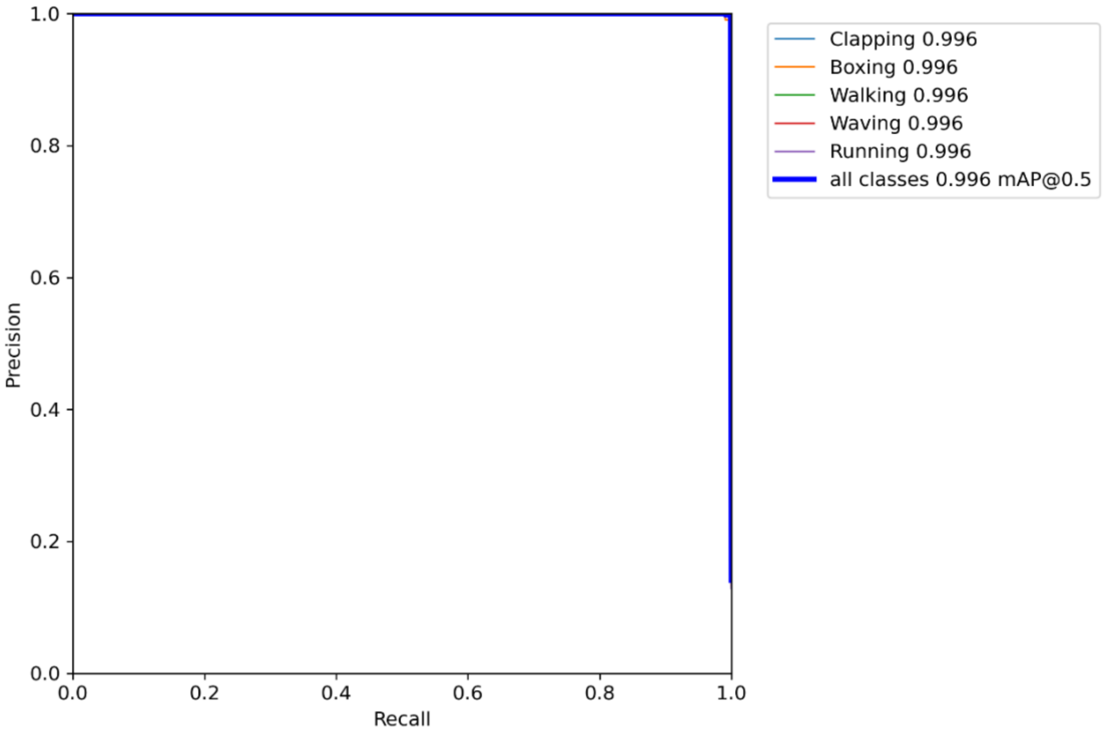


Figure 4.15: PR_curve of YOLOv7+SimAM

For the YOLOv7+CBAM+SimAM network, it costs 1.44 hours for the network to complete 150 epochs of training. Figure 4.16 shows the trend of precision and recall. In the 150 epochs of YOLOv7+SimAM training, both tend to stabilize as the number of epochs enhances. Figure 4.17 shows the changes in mAP values in the YOLOv7+CBAM+SimAM model.

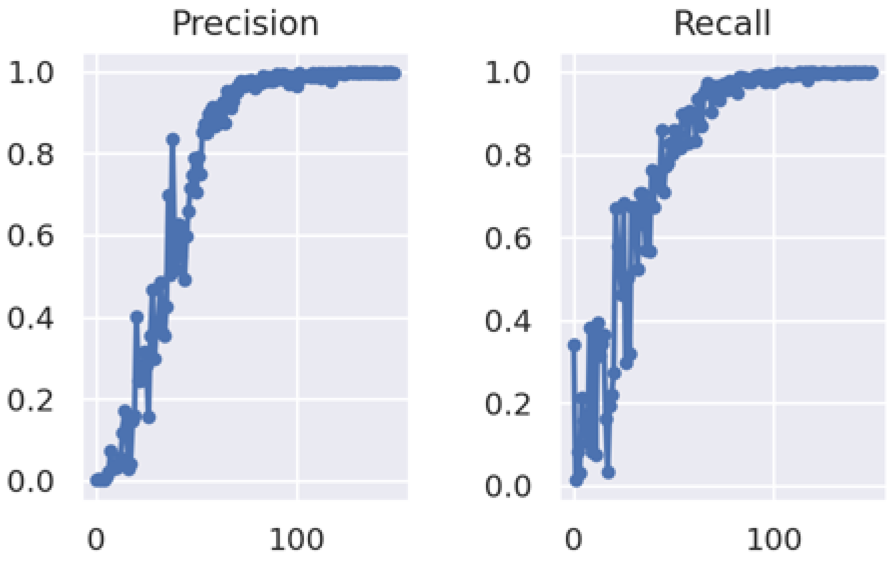


Figure 4.16: The precision and recall of YOLOv7+CBAM+SimAM

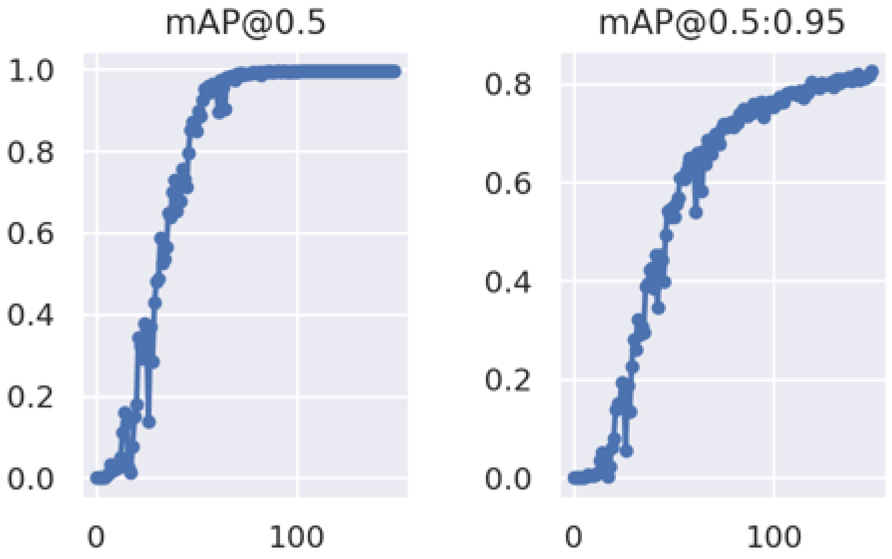


Figure 4.17: The mAP of YOLOv7+CBAM+SimAM

Figure 4.18 shows the relationship between accuracy F1 and confidence of YOLOv7+ CBAM+SimAM. The F1 score ranges from 0 to 1. 1 is the best and 0 is the worst. F1 of YOLOv7+CBAM+SimAM achieves better results with a confidence interval of 0.4 to 0.8. The highest F1 for YOLOv7+CBAM+SimAM is 1.

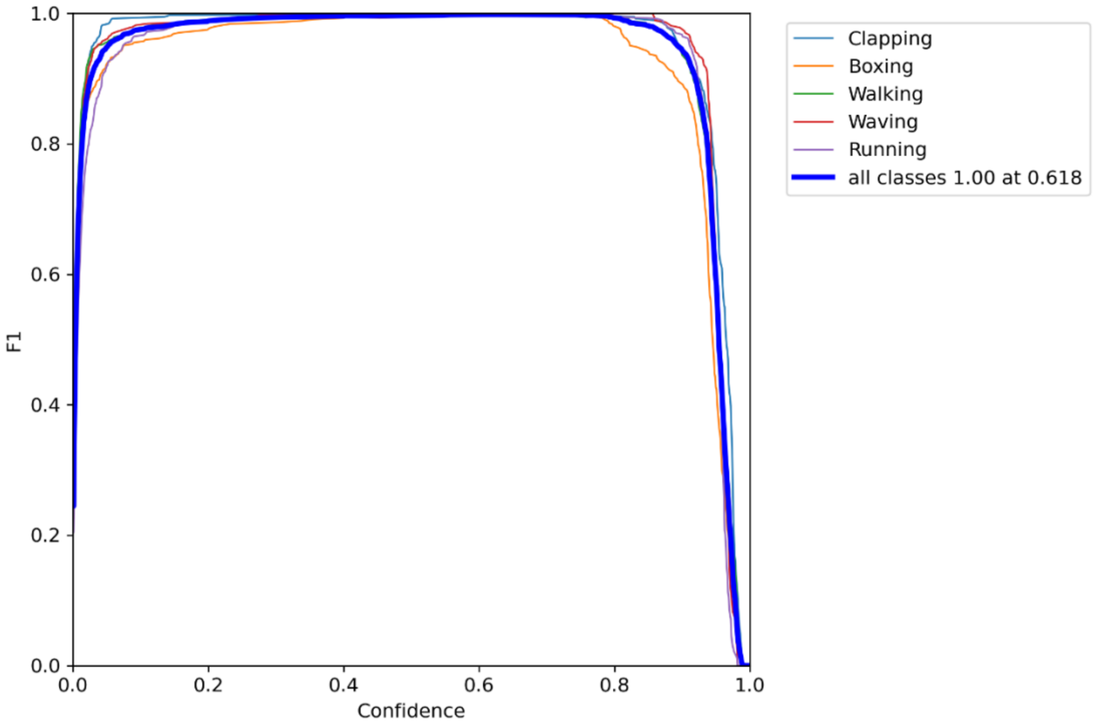


Figure 4.18: F1_curve of YOLOv7+CBAM+SimAM

Figure 4.19 shows the confusion matrix of YOLOv7+CBAM+SimAM, except for the accuracy of action Walking, which is 0.99, the accuracy of all other actions reached 1.00.

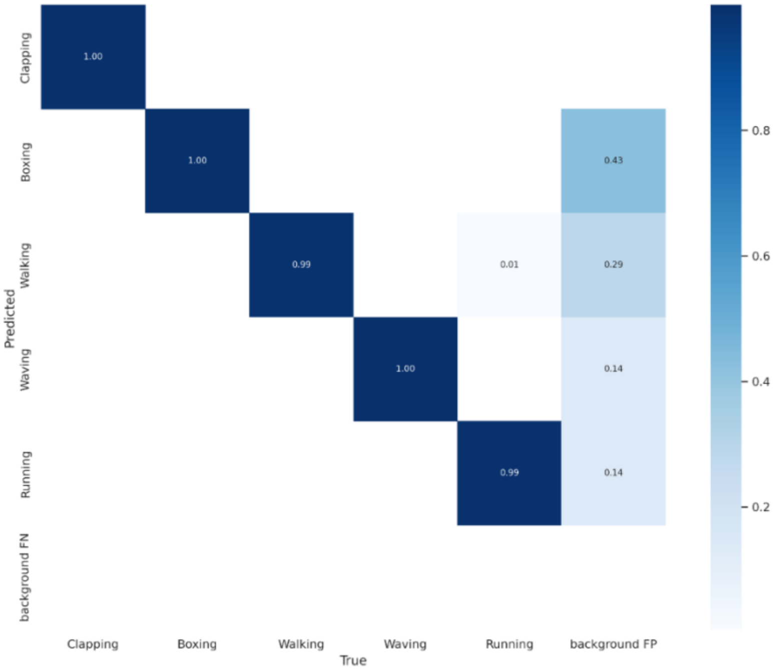


Figure 4.19: The confusion matrix of YOLOv7+CBAM+SimAM

Figure 4.20 shows the PR diagram of YOLOv7+CBAM+SimAM, which represents the relationship between precision and recall. The actions “Clapping”, “Boxing”, “Waving”, “Walking”, and “Running” are all 0.996 for mAP@0.5, which is as same as the overall result for mAP@0.5.

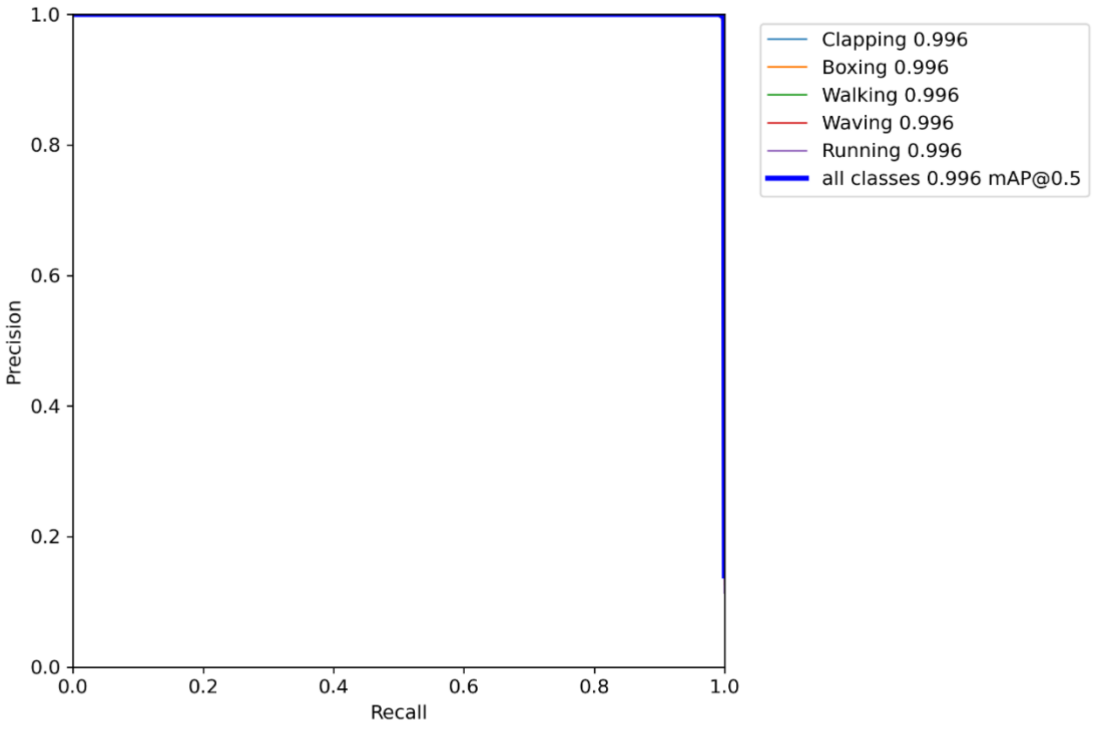
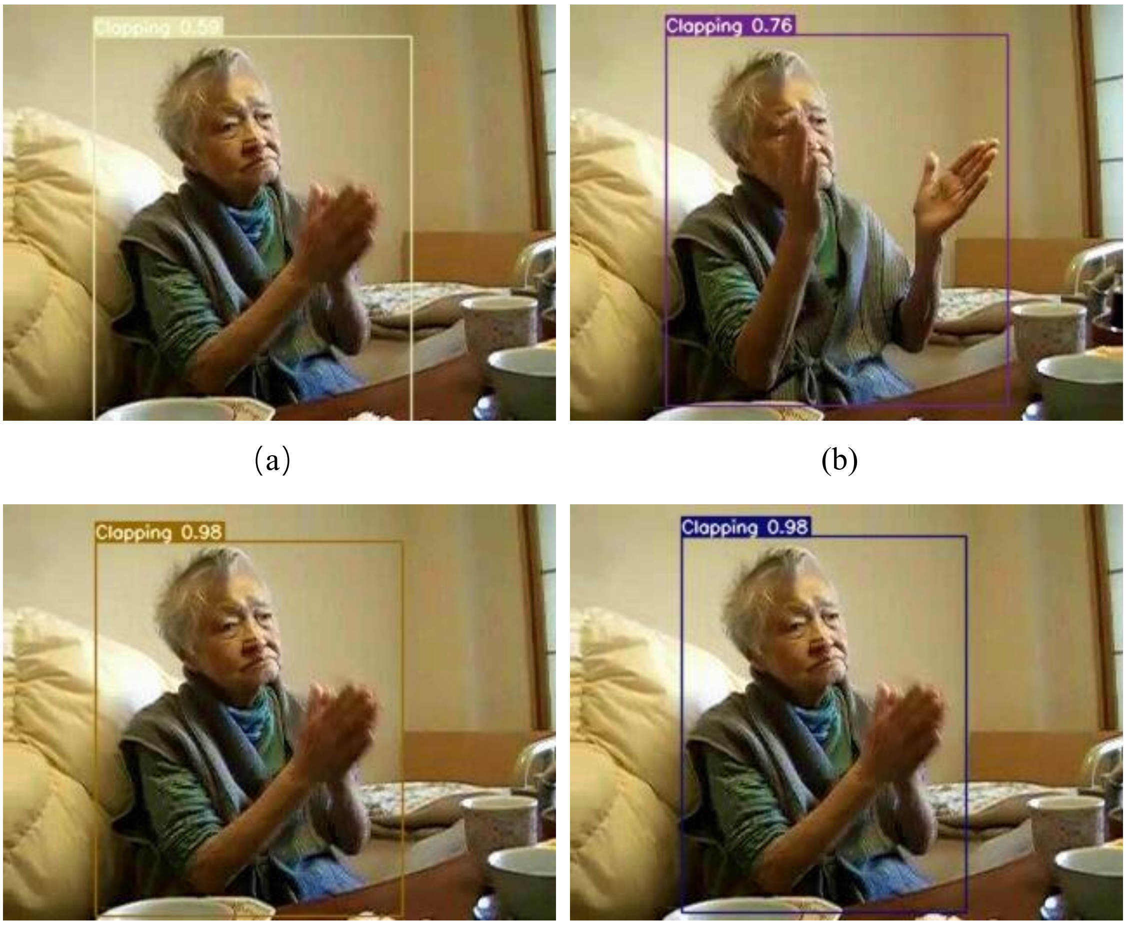


Figure 4.20: PR_curve of YOLOv7+CBAM+SimAM

4.1.2 Human Action Recognition Test Result

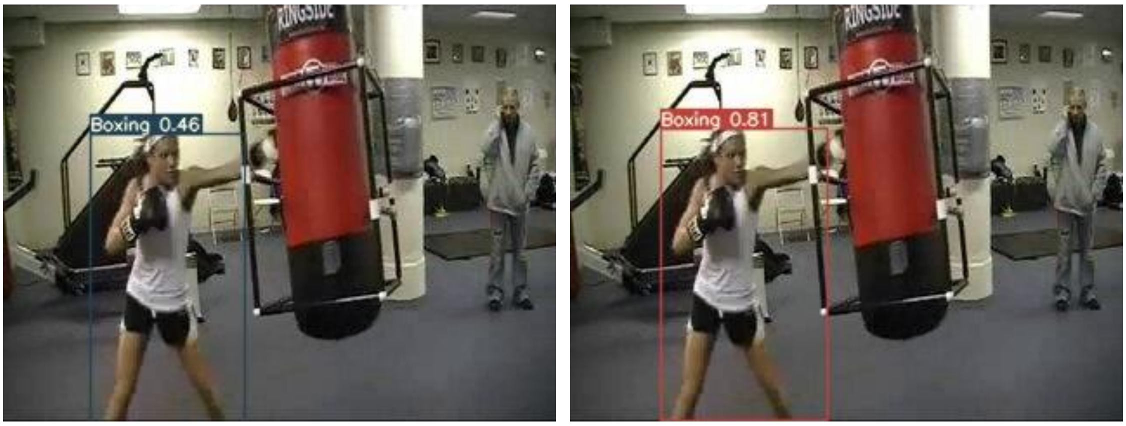
After the training process is complete, it is time to test the performance of the model. Below we show the results of the four models YOLOv7, YOLOv7+CBAM, YOLOv7+SimAM, YOLOV7+CBAM+SimAM after being tested on the test set. The results incorporate the recognition results of but one action, the recognition results of multiple actions and some wrong recognition results. Figure 4.21, Figure 4.22, Figure 4.23, Figure 4.24, and Figure 4.25 show the recognition results of each of the four models for a single action. Image 4.26 shows the recognition results of the four models for multiple actions. We clearly see that the recognition results using the YOLOv7+CBAM+SimAM model are better than the other models. Figure 4.27, on the other hand, shows wrong recognition results. These results are incorrect action recognition, while others are failure to recognize the action.



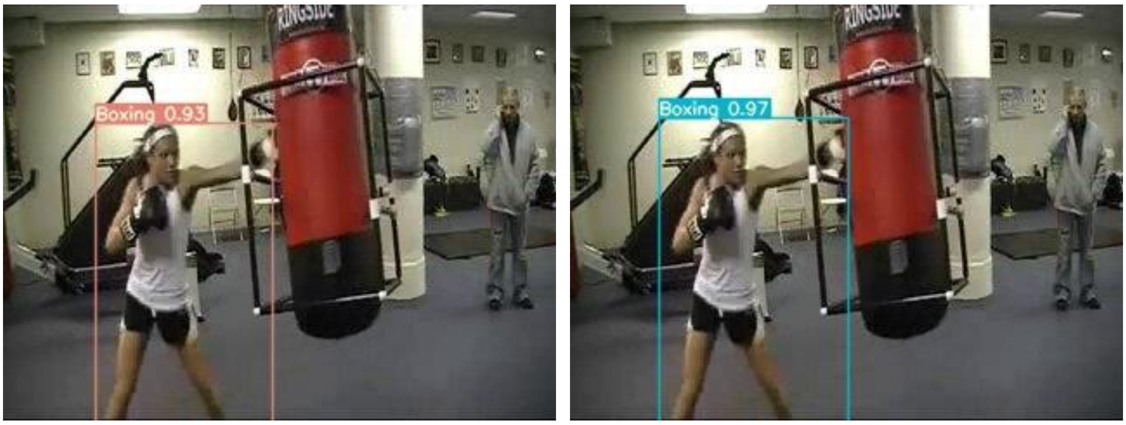
(a) (b)

(c) (d)

Figure 4.21: The results of four models for recognizing action Clapping. (a) YOLOv7, (b) YOLOv7+CBAM, (c)YOLOv7+CBAM, (d)YOLOv7+CBAM+SimAM

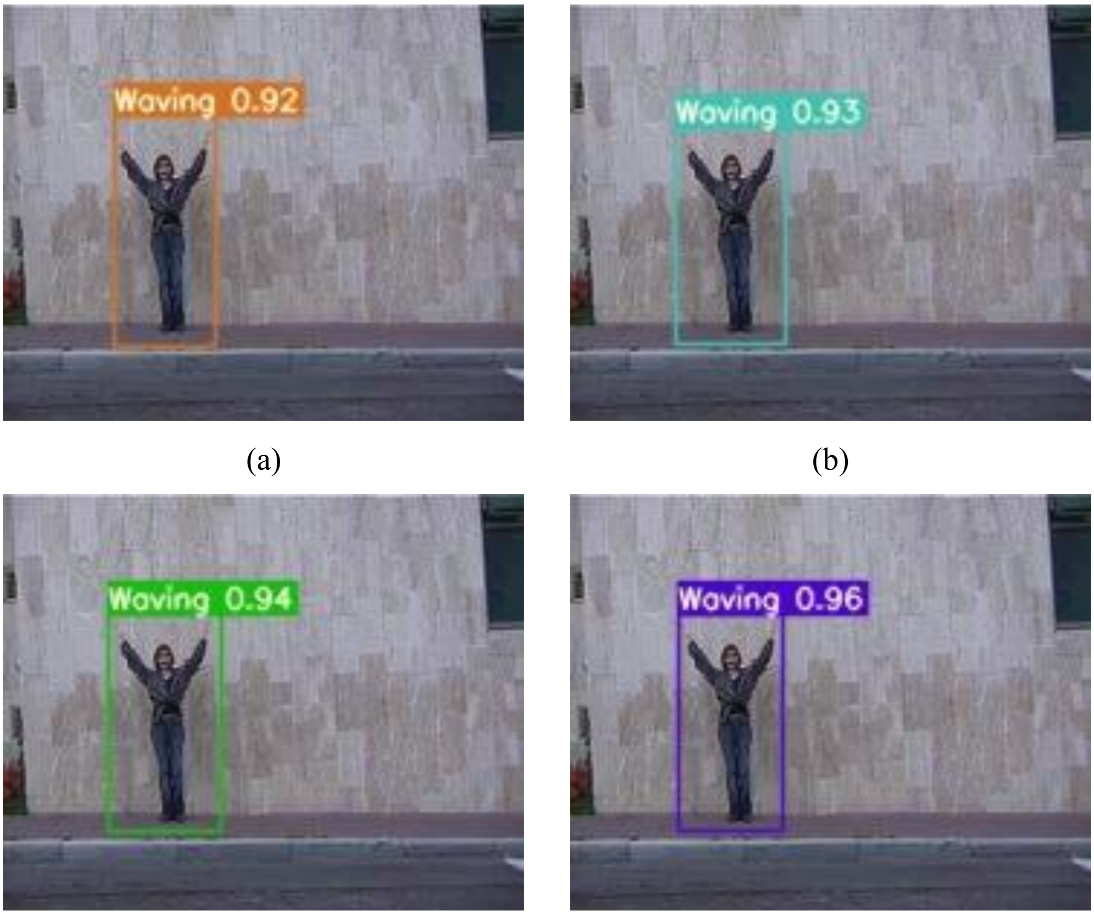


(a) (b)



(c) (d)

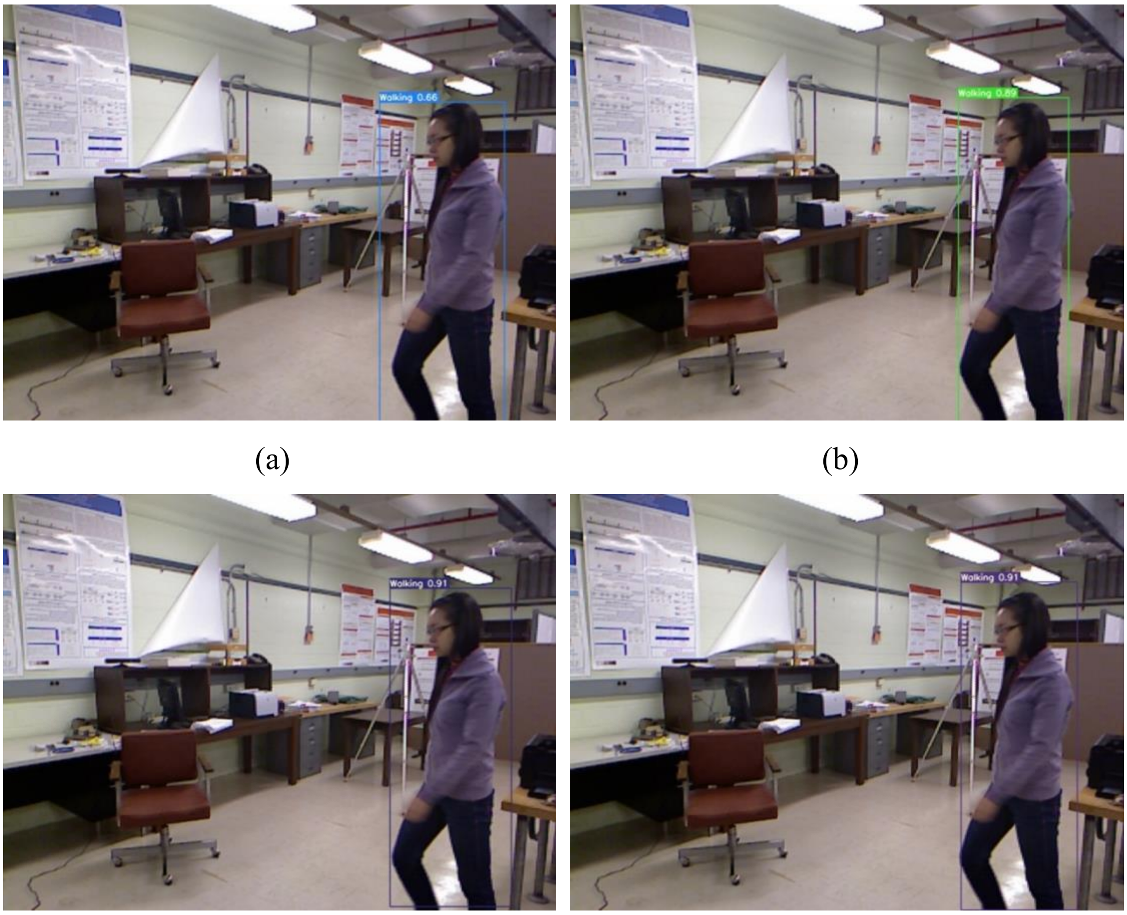
Figure 4.22: The results of four models for recognizing action Boxing. (a) YOLOv7, (b) YOLOv7+CBAM, (c)YOLOv7+CBAM, (d)YOLOv7+CBAM+SimAM



(a) (b)

(c) (d)

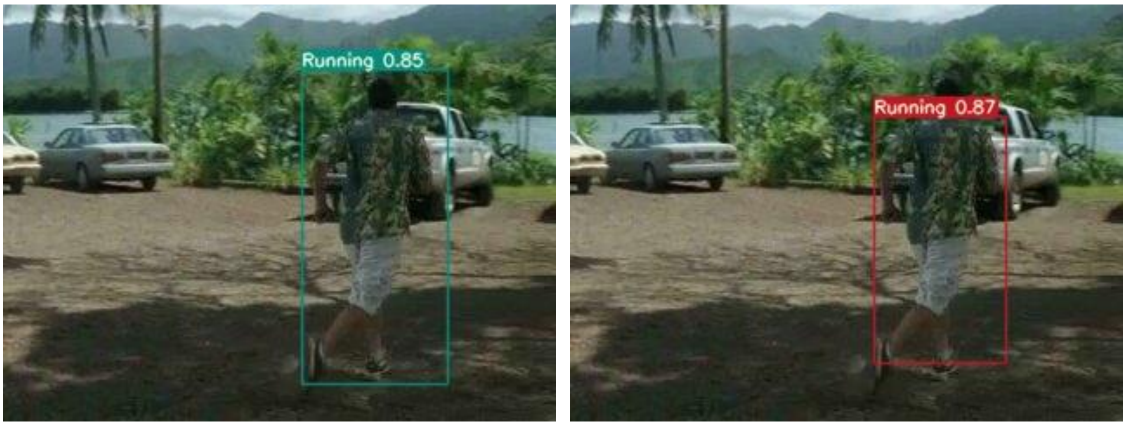
Figure 4.23: The results of 4 models for recognizing action Waving. (a) YOLOv7, (b) YOLOv7+CBAM, (c)YOLOv7+CBAM, (d)YOLOv7+CBAM+SimAM



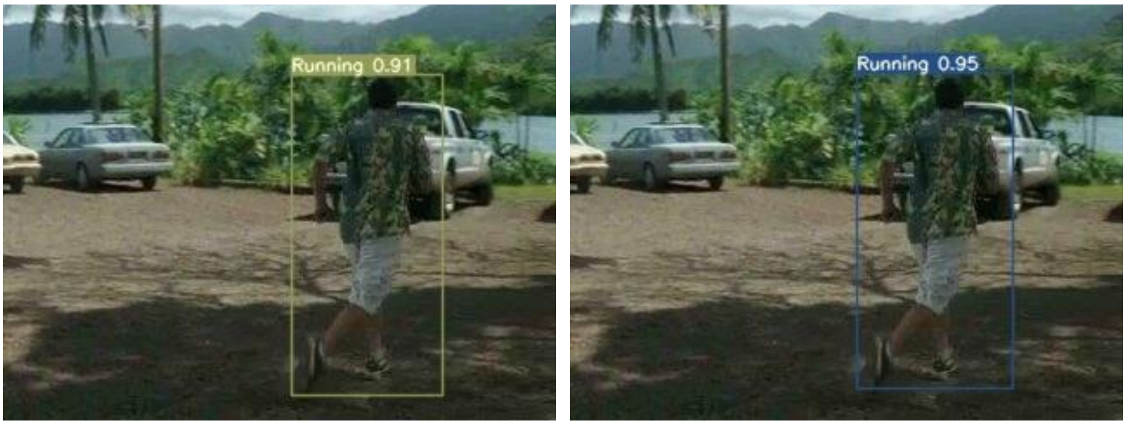
(a) (b)

(c) (d)

Figure 4.24: The results of four models for recognizing action Walking. (a) YOLOv7, (b) YOLOv7+CBAM, (c)YOLOv7+CBAM, (d)YOLOv7+CBAM+SimAM

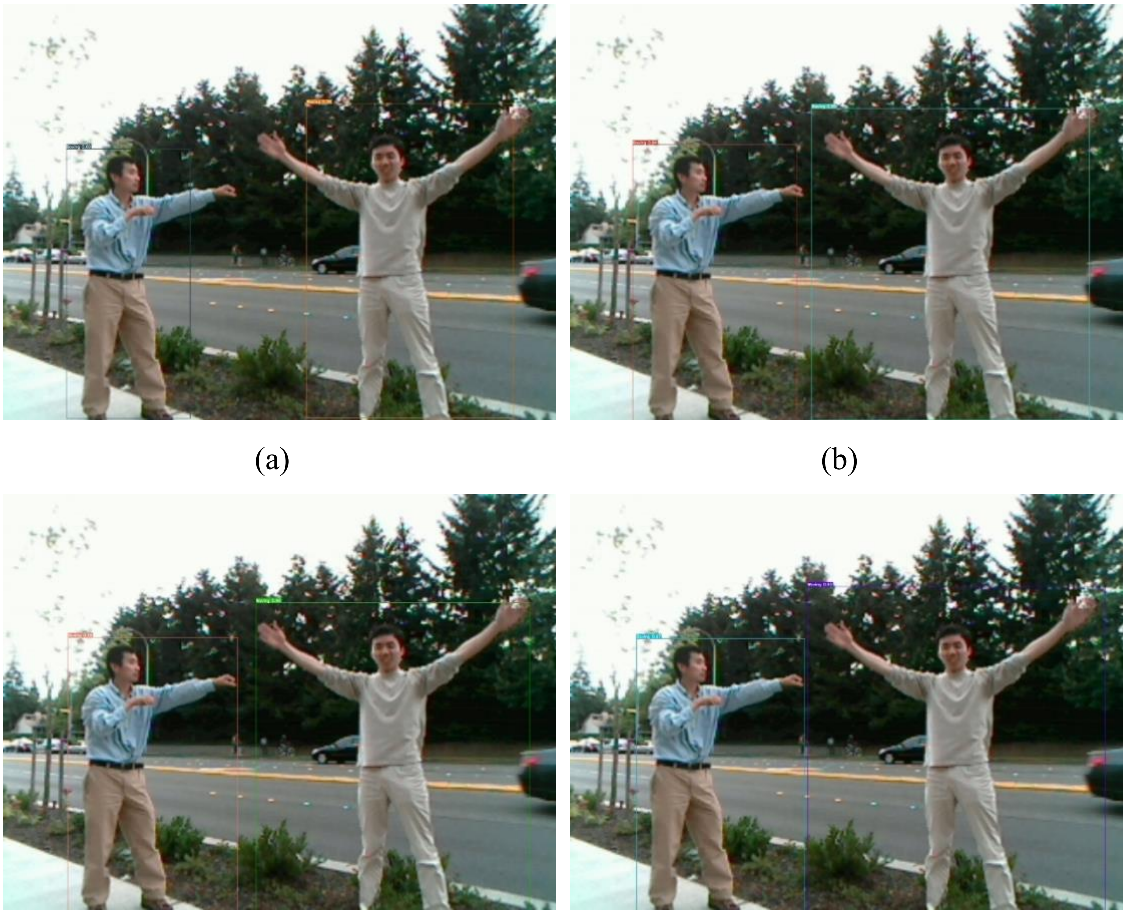


(a) (b)



(c) (d)

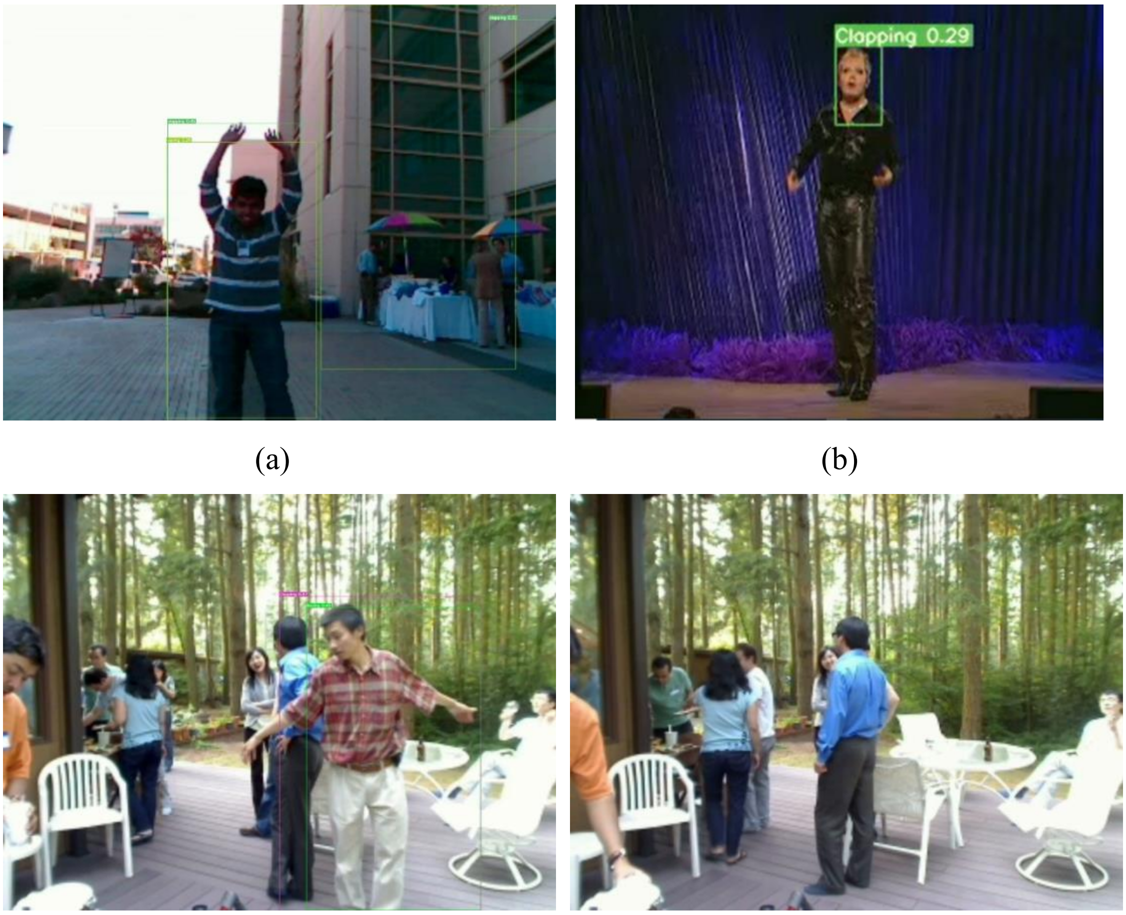
Figure 4.25: The result of four models for recognizing action Running. (a) YOLOv7, (b) YOLOv7+CBAM, (c)YOLOv7+CBAM, (d)YOLOv7+CBAM+SimAM



(a) (b)

(c) (d)

Figure 4.26: The results of two actions recognition using the 4 models. (a) YOLOv7, (b) YOLOv7+CBAM (c)YOLOv7+CBAM, (d)YOLOv7+CBAM+SimAM



(a)

(b)

(c)

(d)

Figure 4.27: The wrong results. (a) Lot wrong label, (b) Label does not completely cover the object, (c) Incorrect recognition result, (d) Recognition fail

4.2 Limitations of the Research

From the experimental results, our proposed algorithm achieves successful outcomes in the human action recognition task. But there are also many limitations that need to be improved. These restrictions include:

Although in this dataset, we collected human actions occurring in disparate contexts and environments from multiple public datasets, there are still some unrecognized or misrecognized actions. There are still some issues with the dataset we created. More context and actions in the environment are needed to optimize the dataset.

Only five classes of human actions can be successfully recognized. In human action related video data, there is more than one action, and often multiple actions appear one after another. Multiple consecutive actions performed in longer videos are not well recognized. Therefore, we need to add more actions within the identifiable range. Expand the number of actions in the dataset.

Also, there is a slight shortage of video data to perform multiple actions simultaneously. Although most of the recognitions are successful, misrecognition occurs, and the recognition result is not good. This shows that the model has not been trained enough due to data problems in the recognition of multiple actions.

Chapter 5
Analysis and Discussions

The focus of this chapter is on analysis and comparison of the various experimental results. The results were from four different YOLO models and will be compared. Specifically, more details on performance comparisons are presented in this chapter.

5.1 Analysis

We directly presented multiple results for human action recognition using YOLO-based models in the previous chapter. We analyze and discuss the results presented in the previous chapters in detail in this chapter. The results that cannot be displayed directly are tabulated for analysis.

The result of loss functions often indicates whether the network performance is superior or not. After a sample passes through the model, a predicted value is obtained. The difference between the predicted value and the true value is the loss. In the process of model training, the smaller the loss, the better the performance of the model. The loss function used by the four models of this thesis is consistent. After compared the last 10 losses of the four models, we found that the loss functions of the four models finally converged. Among the four models, the loss of the network using the attention mechanism is analogous, and the loss of the three networks using the attention mechanism is smaller than that of YOLOv7. A comparison of the most recent 10 epoch losses is shown in Figure 5.1.

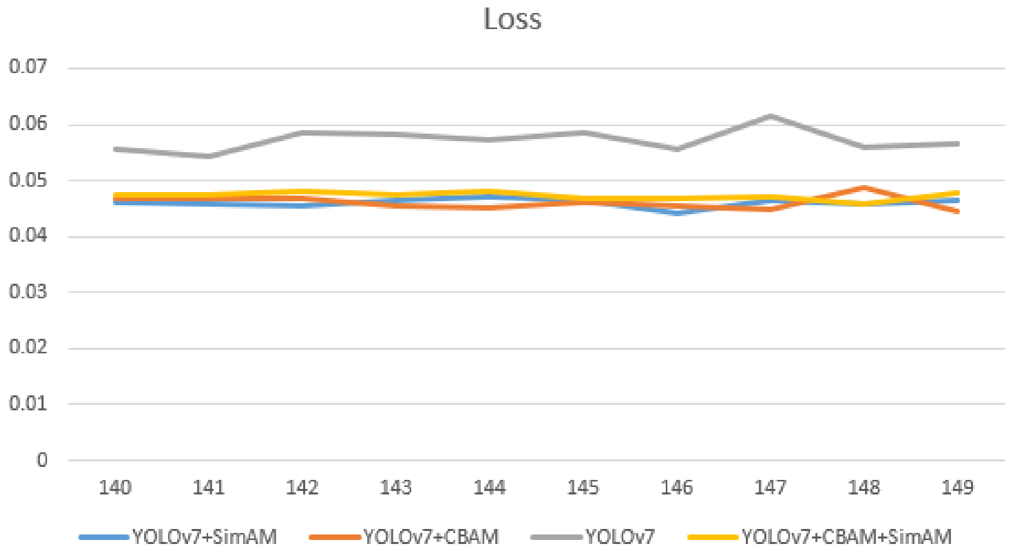
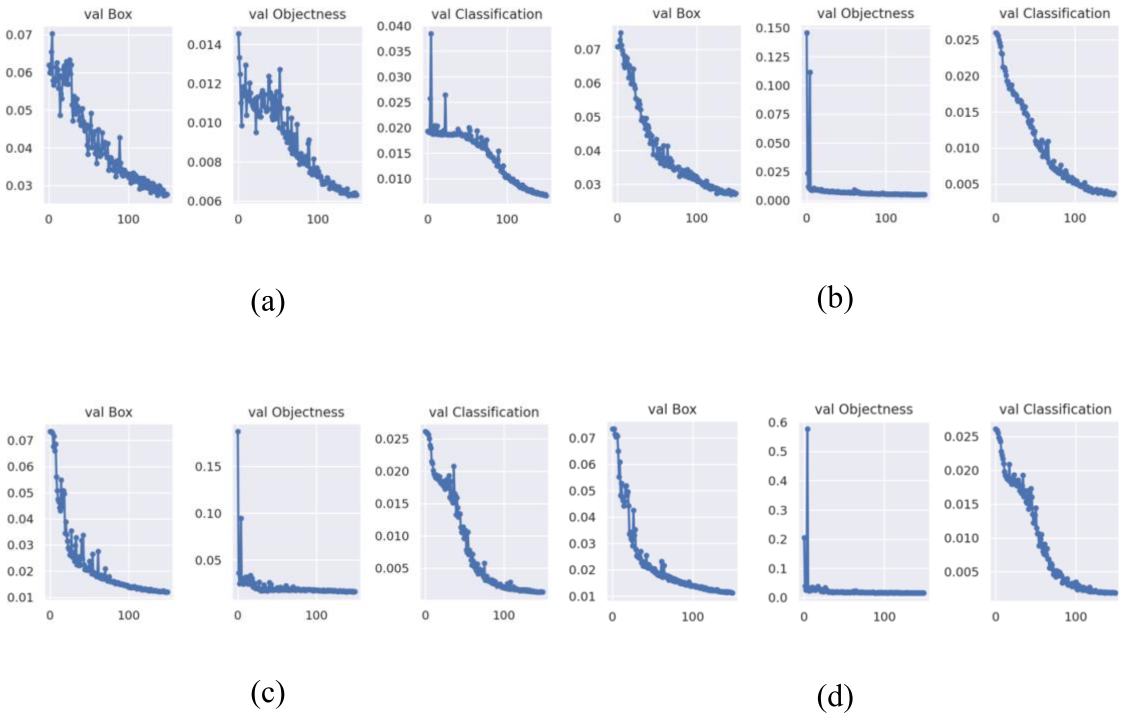


Figure 5.1: The last 10 epoch losses.

The loss on the validation set is equally important. By comparing the validation set losses of the four models, we found that the losses of the four models gradually decreased with the progress of epochs, are finally stabilized. This shows that the four models have been well trained, and the performance of the models is guaranteed. Specifically, the YOLOv7 model has an overall more volatile loss and slightly higher loss values than the other models. Figure 5.2 illustrates the val loss of the four models.



(a)

(b)

(c)

(d)

Figure 5.2: The val loss. (a) YOLOv7, (b) YOLOv7+CBAM (c)YOLOv7+CBAM, (d)YOLOv7+CBAM+SimAM

Besides the losses, there are other ways to evaluate the performance of the model, such as accuracy, precision, recall, mAP, training time. Among them, except for the training time, other four values are bigger. Table 5.1 shows the similarities and differences of these metrics across the four models. Among them, the four indicators of YOLOv7 are the worst among the four models. In terms of running time, YOLOv7+SimAM has the shortest running time, which is 1.438 hours. But on the other four indicators, YOLOv7+CBAM+SimAM is either higher than other models or is numerically equal to other models. For example, YOLOv7+CBAM+SimAM is equal to YOLOv7+SimAM in terms of accuracy, and 0.08 higher than YOLOv7+CBAM.

Table 5.1: The performance comparison.

Model	**Accuracy**	**Precision**	**Recall**	**mAP @.5**	**Training time**
YOLOv7	0.930	0.933	0.912	0.970	1.500 hours
YOLOv7+CBAM	0.990	0.990	0.993	0.995	1.470 hours
YOLOv7+SimAM	0.994	0.998	0.997	0.996	1.438 hours
YOLOv7+CBAM+SimAM	0.996	0.998	0.999	0.996	1.441 hours

PR_cruve refers specifically to the RP graph. Among them, *P* represents the precision, and *R* refers to the recall rate. The changes in the graph represent the relationship between precision and recall. The RP curve is generated with recall as the abscissa and precision as the ordinate. The area enclosed under the curve can be identified as AP. The area enclosed by the PR curve is AP. In a PR diagram, if one curve *A* in the PR diagram completely surrounds the other curve *B*. Then, *A* must be better than *B* at this time. The closer the curve of the image is to the upper right corner, the better the performance of the model represented by the curve. PR cruve generated by observing the four models. We found that the curves generated by all models did not fluctuate much, indicating that the training effect was good. In the areas generated under the curve, only YOLOv7 model has a significantly smaller area than the other models, only 97% of the total area. The area enclosed by other models is YOLOv7+CBAM, equals to 99.5% of the total area. YOLOv7+SimAM for 99.6% of the total area. YOLOv7+CBAM+SimAM for 99.6% of the total area. This shows that the training effect of the three is basically the same. The training effect of YOLOv7 is slightly worse. Figure 5.3 shows the PR map comparison of the four models.

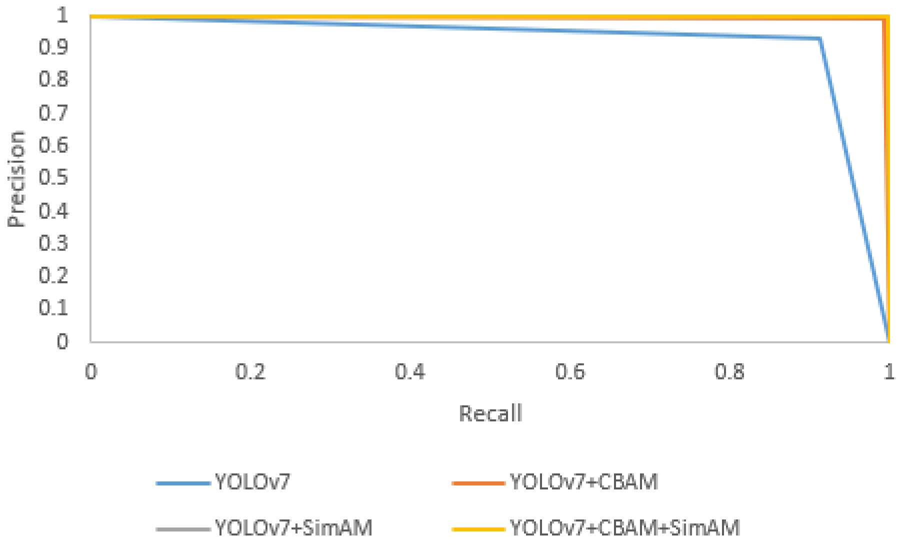


Figure 5.3: The PR of four models

In addition, the results of individual actions are also noteworthy. By comparing numerical values such as the accuracy of individual actions in disparate models, it is possible to intuitively understand which actions are well-recognized in disparate models. If an action needs to be recognized, it is clear which model is better at recognizing that action.

Table 5.2: The final result of each action in YOLOv7

Action	**Accuracy**	**Precision**	**Recall**	**mAP @.5**
Clapping	0.950	0.889	0.947	0.980
Boxing	0.900	0.966	0.797	0.964
Walking	0.920	0.902	0.952	0.970
Waving	0.940	0.925	0.936	0.973
Running	0.940	0.983	0.930	0.963

As shown in Table 5.2, by using YOLOv7, the “Running” action is the one with higher values among all actions. The recognition of the clapping action is the shortcoming. This is because the accuracy of clapping is too low compared to other actions. Overall, YOLO without attention mechanism is still inferior to YOLO with attention mechanism in action recognition.

Table 5.3: The final result of each action in YOLOv7+CBAM

Action	**Accuracy**	**Precision**	**Recall**	**mAP @.5**
Clapping	0.990	0.995	0.997	0.995
Boxing	0.990	0.986	0.984	0.995
Walking	0.980	0.981	0.996	0.995
Waving	1.000	0.989	1.000	0.973
Running	0.990	1.000	0.987	0.995

From Table 5.3, we get that action “Waving” is the best action that YOLO+CBAM is good at recognizing. Among them, recall and accuracy reached 1.0. In addition, the recognition of running also performed well. Its precision also reaches 1.0.

Table 5.4: The final result of each action in YOLOv7+SimAM

Action	**Accuracy**	**Precision**	**Recall**	**mAP @.5**
Clapping	1.00	0.997	1.000	0.996
Boxing	0.990	1.000	0.983	0.996
Walking	0.980	0.996	1.000	0.996
Waving	1.000	1.000	1.000	0.996
Running	1.000	0.998	1.000	0.996

The action with the highest recognition accuracy on YOLO+SimAM is also the action “Waving”. All values are 1.0 except mAP. The evaluation metrics for other actions were equally high. In terms of accuracy, “Running”, “Waving”, and “Clapping” all have reached 100%. The mAP values are 0.996, which further indicates that all actions in the model are well recognized.

Table 5.5: The final result of each action in YOLOv7+CBAM+SimAM

Action	**Accuracy**	**Precision**	**Recall**	**mAP @.5**
Clapping	1.000	0.999	1.000	0.996
Boxing	1.000	1.000	0.995	0.996
Walking	0.990	0.991	1.000	0.996
Waving	1.000	0.998	1.000	0.996
Running	0.990	1.000	0.999	0.996

The performance based on YOLO+CBAM+SimAM model is much better. As shown in Table 5.5, the accuracy, precision, recall, and mAP all reached 0.99. Many of them even reached the level of 1. In general, the performance of YOLO+CBAM+SimAM and YOLO+SimAM is not much different. The performance of single action recognition is very good.

Disparate models based on the test set also have differences in recognition speed. Since the test set consists of video and image together. So, the time is also split into two parts, 6 videos consist of 1,364 frames. The test set contains a total of 1,494 frames.

Table 5.6: The differences in detection time for various models.

	YOLOv7	YOLOv7+ CBAM	YOLOv7+ SimAM	YOLOv7+ CBAM+SimAM
Image data detection time (s)	73.030s	71.930s	40.197s	40.647s
Video data detection time (s)	764.723s	731.655s	399.864s	422.835s
Detection time spent per frame (ms)	561ms	537ms	295ms	310ms

The final results are shown in Table 5.6, YOLOv7+ SimAM has the fastest processing speed, followed by YOLOv7+ CBAM+SimAM.

5.2 Discussions

We experimentally compared four disparate models. The accuracy rate of human action recognition through the YOLOv7 model is 93%, the accuracy rate is 93%, the recall rate is 0.912, the mAP@.5 is 0.970, and the training time is 1.5 hours. The recognition time of a single frame is 561ms. Compared to the other models, the action recognition results using only YOLOv7 are relatively poor among the four models. However, the overall recognition accuracy still maintains a good level, and it is also a successful human action recognition model.

Compared with YOLOv7, the YOLOv7+CBAM model has indeed been improved in terms of accuracy, precision, and other indicators. Among them, the accuracy is 99%, the precision is 99%, the recall is 0.993, and the mAP@.5 is 0.995. But the running time of this model is not much faster than YOLOv7. The training time of the model is 1.47 hours, which is only 0.3 hours faster than the base model. The recognition time for a single frame is 537ms, which is only 24ms faster than the base model. In general, though the overall recognition has reached a good level, there are still some shortcomings in single-action recognition, such as boxing.

The YOLOv7+SimAM model is another successful model. In the experiment, its indicators are mostly among the best among the four models. The overall accuracy of the model is 99.4%, the precision is 99.8%, the recall is 0.997, and the mAP@.5 is 0.996. In terms of training time, this model is the fastest of the four models, taking only 1.438 hours. The recognition time of a single frame is also the fastest at 295ms. The accuracy also reaches 99% in the recognition of a single action, which is a very good model.

YOLOv7+CBAM+SimAM model is based on the YOLOv7+CBAM model and YOLOv7+SimAM, adding two attention mechanisms to the network to jointly perform the task of human action recognition. In terms of accuracy, the model achieved an accuracy of 99.6%. In addition, the precision is 99.8%, the recall is 0.999, and the mAP@.5 is 0.996. The running time is also greatly reduced compared to YOLOv7. The detection time of a single frame is only 310ms. However, the training time of YOLOv7+CBAM+SimAM is slightly longer than that of the YOLOv7+SimAM model. The training time of the model is 1.441 hours, which is 0.4 hours slower than YOLOv7+SimAM. This may be due to the two attention mechanisms used in the network. In the recognition of a single action, the performance of the model is also excellent, and the recognition performance of all actions is higher than that using the YOLOv7+SimAM model.

In conclusion, our deep learning methods based on the YOLO model and attention mechanism can accurately identify human actions in videos. In addition, the running speed of the model also has certain advantages. The implementation of three disparate model methods improves the accuracy of action recognition using YOLO models. The proposed four models all have good recognition performance. Overall, the task of recognizing human actions has been successfully accomplished.

Chapter 6
Conclusion and Future Work

In this chapter, we firstly give a general summary of the whole thesis. We elaborate on the thesis and methods. Then, based on the results and shortcomings of the experiments, we find future research directions, we summarize the work that needs to be conducted in the future.

6.1 Conclusion

The goal of this thesis is to find ways to effectively identify human actions. After summarizing human action recognition methods, we propose a YOLO-based approach for human action recognition. The latest YOLOv7 for human action recognition was proposed. In addition, two disparate attention mechanisms were employed based on YOLOv7. These two attention mechanisms were successfully applied to YOLOv7 for human action recognition, and three YOLO-attention models were implemented, namely YOLOv7+CBAM, YOLOv7+SimAM and YOLOv7+CBAM+SimAM models.

In this thesis, we successfully demonstrate that it is feasible to use attention mechanism based on YOLOv7 for human action recognition. The experimental results show that the accuracy of YOLOv7 using the attention mechanism is improved up to 7%. The YOLOv7+CBAM, YOLOv7+SimAM and YOLOv7+CBAM+SimAM models proposed in this thesis can all achieve good accuracy results, 99%, 99.4% and 99.6%, respectively. The accuracy of YOLOv7+CBAM+SimAM model is the highest one. The running time is faster than the base YOLOv7 model. Also, despite the runtime of YOLOv7+SimAM is faster, only 1.5% faster than the YOLOv7+CBAM+SimAM model. The accuracy of the YOLOv7+CBAM+SimAM model is 0.2% higher than that of the YOLOv7+SimAM model. Therefore, among all the models proposed in this thesis, the YOLOv7+CBAM+SimAM model performs the best.

6.2 Future Work

In future, YOLOv7 can continue being served as the network foundation. On this basis, various networks are continuously developed in YOLO. A better model compared to the current model is sought.

Using more efficient convolutional layers to replace the existing generic convolutions is our future work direction. In addition, the attention mechanism in YOLOv7 further reduces the running time of YOLOv7+CBAM+SimAM. We will load more models in

YOLOv7 to improve the running speed and recognition effect.

Another work that needs to be implemented is to expand the datasets. The public datasets were developed based on the datasets we create. We will increase more classes of human actions in the datasets with more complex backgrounds. A larger and richer dataset will improve the final result of model training. In addition, human action recognition in real time is also a working direction for us.

References

Ali, V. S. (2017). Deep Temporal Linear Encoding Networks. *Computer Vision and Pattern Recognition*, 2329-2338.

AminUllaha, K., UlHaqa, I., & WookBaik, S. (2019). Action Recognition Using Optimized Deep Autoencoder and CNN for Surveillance Data streams of Non-stationary Environments. *Future Generation Computer Systems*, 386-397.

An, N., & Yan, W. Q. (2021). Multitarget Tracking Using Siamese Neural Networks. *ACM Transactions on Multimidia Computing Communications and Applications*, 1-16.

Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., & Schmid, C. (2021). ViViT: A Video Vision Transformer. *International Conference on Computer Vision*, 6836-6846.

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. arXiv:1409.0473.

Bansal, M., Yan, W.-Q., & Kankanhalli, M. (2003). Dynamic Watermarking of Images. *International Conference on Information, Communications and Signal Processing*, pp.965-969.

Bertasius, e., Wang, H., & Torresani, L. (2021). Is Space-Time Attention All You Need for Video Understanding? *ICML*, 4.

Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., . . . Ghayvat, H. (2021). CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope. *Electronics*, 2470.

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv:2004.10934.

Cao, X. (2022) Pose Estimation of Swimmers from Digital Images Using Deep Learning.

Master's Thesis, Auckland University of Technology.

Carreira, J., & Zisserman, A. (2017). Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. *Conference on Computer Vision and Pattern Recognition* , 6299-6308.

Chauhan, R., Ghanshala, K. K., & Joshi, R. (2018). Convolutional Neural Network (CNN) for Image Detection and Recognition. *International Conference on Secure Cyber Computing and Communication*, 278-282.

Chien-Yao Wang, H.-Y. M.-H., Chen, P.-Y., Hsieh, J.-W., & Yeh, I.-H. (2020). CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. *Computer Vision and Pattern Recognition* , 390-391.

Chuyi Li, L. L., Chu, X., Wei, X., & Wei, X. (2022). YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. arXiv:2209.02976.

Cui, W., Yan, W. (2016) A scheme for face recognition in complex environments. International Journal of Digital Crime and Forensics (IJDCF) 8 (1), 26-36.

Diba, A., Fayyaz, M., Sharma, V., Karami, A. H., Arzani, M. M., Yousefzadeh, R., & Gool, L. V. (2017). Temporal 3D ConvNets: New Architecture and Transfer Learning for Video Classification. arXiv:1711.08200.

DiPietro, R., & D.Hager, G. (2020). Deep learning: RNNs and LSTM. In *Handbook of Medical Image Computing and Computer Assisted Intervention* (pp. 503-519). Academic Press.

Donahue, J., Hendricks, L. A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Darrell, T., & Saenko, K. (2015). Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. *Conference on Computer Vision and Pattern Recognition*, 2625-2634.

Du, W., Wang, Y., & Qiao, Y. (2017). RPAN: An End-To-End Recurrent Pose-Attention Network for Action Recognition in Videos. *International Conference on Computer Vision*, 3725-3734.

Everingham, M., Eslami, S. M., Gool, L. V., Williams, C. K., Winn, J., & Zisserman, A. (2015). The PASCAL Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision*, 93-136.

Fan, C.-S., Liang, J.-M., Lin, Y.-T., Wu, K.-R., Li, K.-Y., Lin, T.-Y., & Tseng, Y.-C. (2015). A Survey of Intelligent Video Surveillance Systems: History, Applications and Future. *Frontiers in Artificial Intelligence and Applications*, 1479-1488.

Fan, Q., Chen, C.-F., & Panda, R. (2021). Can an Image Classifier Suffice For Action Recognition? *International Conference on Learning Representations*, 2021.

Feichtenhofer, C., Pinz, A., & Zisserman, A. (2016). Convolutional Two-Stream Network Fusion for Video Action Recognition. *Computer Vision and Pattern Recognition*, 1933-1941.

Gai, R., Chen, N., & Yuan, H. (2021). A Detection Algorithm for Cherry Fruits Based on the Improved YOLO-v4 model. *Neural Computing and Applications*, 1-12.

Gammulle, H., Denman, S., Sridharan, S., & Fookes, C. (2017). Two Stream LSTM: A Deep Fusion Framework for Human Action Recognition. *Winter Conference on Applications of Computer Vision*, 177-186.

Ge, Z., Liu, S., Wang, F., Li, Z., & Sun, J. (2021). YOLOX: Exceeding YOLO Series in 2021. arXiv:2107.08430.

Gorelick, L., Blank, M., Shechtman, E., Irani, M., & Basri, R. (2005). Actions as Space-Time Shapes. *International Conference on Computer Vision*, pp. 1395-1402.

Grasso, C., & Schembra, G. (2018). Design of a UAV-Based Videosurveillance System with Tactile Internet Constraints in A 5G Ecosystem. *IEEE Conference on*

Network Softwarization and Workshops (NetSoft), 449-455.

Gul, M. A., Yousaf, M. H., Nawaz, S., Rehman, Z. U., & Kim, H. (2020). Patient Monitoring by Abnormal Human Activity Recognition Based on CNN Architecture. *Electronics*, 1993.

Gulzar, N., Abbasi, B., Wu, E., Ozbal, A., & Yan, W. (2013). Surveillance Privacy Protection. *Intelligent Multimedia Surveillance*, pp.83-105.

Guo, J., Mu, Y., Xiong, M., Liu, Y., & Gu, J. (2019). Complex Methods Applied to Data Analysis, Processing, and Visualization. *Complexity*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Computer Vision and Pattern Recognition*, 770-778.

Herrera, A., Beck, A., Bell, D., Miller, P., Wu, Q., Yan, W. (2008) Behavior Analysis and Prediction in Image Sequences Using Rough Sets. International Machine Vision and Image Processing Conference (pp.71-76)

Horn, B. K., & Rhunck, B. G. (1981). Determining Optical Flow. *Artificial intelligence*, 185-203.

Hussain, A., Hussain, T., Ullah, W., & Baik, S. W. (2022). Vision Transformer and Deep Sequence Learning for Human Activity Recognition in Surveillance Videos. *Computational Intelligence and Neuroscience*.

Jaswinder, S., & Banerjee, R. (2019). A Study on Single and Multi-layer Perceptron Neural Network. *International Conference on Computing Methodologies and Communication*, 35-40.

Ji, S., Xu, W., Yang, M., & Yu, K. (2012). 3D Convolutional Neural Networks for Human Action Recognition. *Transactions on Pttern Aalysis and Mchine Itelligence*, 221-231.

Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., & Qu, R. (2019). A Survey of Deep

Learning-Based Object Detection. *IEEE Access*, 128837-128868.

Jinhyung Kim, S. C., Wee, D., Bae, S., & Kim, J. (2020). Regularization on Spatio-Temporally Smoothed Feature for Action Recognition. *Computer Vision and Pattern Recognition*, 12103-12112.

Johansson, G. (1973). Visual Perception of Biological Motion and A Model for Its Analysis. *Perception & Psychophysics*, 201-221.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale Video Classification with Convolutional Neural Networks. *Computer Vision and Pattern Recognition*, 1725-1732.

Khan, M. A., Akram, T., Sharif, M., Muhammad, N., Javed, M. Y., & Naqvi, S. R. (2020). Improved Strategy for Human Action Recognition; Experiencing A Cascaded Design. *IET Image Processing*, 818-829.

Kieran, D., & Yan, W. (2010). A Framework for an Event Driven Video Surveillance System. *IEEE International Conference on Advanced Video and Signal Based Surveillance* , pp.97-102.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*, 84-90.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., & Serre, T. (2011). HMDB: A Large Video Database for Human Motion Recognition. *International Conference on Computer Vision*, 2556-2563.

L Xia, C. C. (2012). View Invariant Human Action Recognition Using Histograms of 3D Joints. *Computer Vision and Pattern Recognition*, 20-27.

Lan, Z., Yu, S.-I., Lin, M., Raj, B., & Hauptmann, A. G. (2015). Handcrafted Local Features are Convolutional Neural Networks. arXiv:1511.05045.

Lan, Z., Zhu, Y., Hauptmann, A. G., & Newsam, S. (2017). Deep Local Video Feature for

Action Recognition. *Conference on Computer Vision and Pattern Recognition*, 1-7.

Le, H., Nguyen, M., Yan, W. Q., & Nguyen, H. (2021). Augmented Reality and Machine Learning Incorporation Using YOLOv3 and ARKit. *Applied Sciences*, 6006.

Lemley, J., Bazrafkan, S., & Corcoran, P. (2017). Deep Learning for Consumer Devices and Services: Pushing the Limits For Machine Learning, Artificial Intelligence, and Computer Vision. *IEEE Consumer Electronics Magazine* , 48-56.

Li, F., Zhang, Y., Yan, W., Klette, R. (2016) Adaptive and Compressive Target Tracking Based on Feature Point Matching. International Conference on Pattern Recognition (ICPR), (pp.2734-2739).

Li, Q., Qiu, Z., Yao, T., Mei, T., Rui, Y., & Luo, J. (2016). Action Recognition by Learning Deep Multi-Granular Spatio-Temporal Video Representation. *ACM on International Conference on Multimedia Retrieval*, 159-166.

Liang, C., LU, J., & YAN, W. Q. (2022). Human Action Recognition From Digital Videos Based on Deep Learning. *ACM ICCCV*.

Li-ming, X., Huang, J.-x., & Tan, L.-z. (2013). Human Action Recognition Based on Chaotic Invariants. *Journal of Central South University*, 3171-3179.

Liu, C., Yan, W. (2020) Gait recognition using deep learning. Handbook of Research on Multimedia Cyber Security (pp.214-226)

Liu, J., Yan, W. (2022) Crime Prediction from Surveillance Videos using Deep Learning. Aiding Forensic Investigation Through Deep Learning and Machine Learning Frameworks. IGI Global

Liu, Z., Yan, W. Q., & Yang, M. L. (2018). Image Denoising Based on A CNN Model. *International Conference on Control, Automation and Robotics*, pp.389-393.

Lu, J., & Yan, W. Q. (2020). Comparative Evaluations of Human Behavior Recognition

Using Deep Learning. In *Handbook of Research on Multimedia Cyber Security* (pp. pp.176-189). IGI Global.

Lu, J., Shen, J., Yan, W. Q., & Bačić, B. (2018). An Empirical Study for Human Behavior Analysis. *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* , pp.1-6.

Lu, J., Yan, W. Q., & Nguyen, M. (2018). Human Behaviour Recognition Using Deep Learning. *Advanced Video and Signal Based Surveillance* , 1-6.

Lu, J., Yan, W. Q., & Nguyen, M. (2020). Deep Learning Methods for Human Behavior Recognition. *Image and Vision Computing New Zealand (IVCNZ)*, 1-6.

Luo, Z., Yan, W. Q., & Nguyen, M. (2022). Kayak and Sailboat Detection Based on the Improved YOLO. *ACM ICCCV*.

Lyon, D. (2003). Surveillance Technology and Surveillance. Modernity and Technology. Emerging Digital Spaces in Contemporary Society, pp 107–120.

Ma, Z. (2021). Human Action Recognition in Smart Cultural Tourism Based on Fusion Techniques of Virtual Reality and SOM Neural Network. *Computational Intelligence and Neuroscience*.

Madhiarasan, M., & Deepa, S. N. (2017). Comparative Analysis on Hidden Neurons Estimation in Multi Layer Perceptron Neural Networks for Wind Speed Forecasting. *Artificial Intelligence Review*, 449-471.

Mail, V. W., & Lucas, T. (2018). Computer Vision and Image Processing: A Paper Review. *International Journal of Artificial Intelligence Research*, 29-36.

Malibari, A. A., Alzahrani, J. S., Qahmash, A., Maray, M., Alghamdi, M., Alshahrani, R., . . . Hilal, A. M. (2022). Quantum Water Strider Algorithm with Hybrid-Deep-Learning-Based Activity Recognition for Human–Computer Interaction. *Applied Sciences*, 6848.

Manish, M., & Srivastava, M. (2014). A View of Artificial Neural Network. *International Conference on Advances in Engineering & Technology Research*, 1-3.

MathieuBarnachon, SaïdaBouakaz, BoubakeurBoufama, & ErwanGuilloua. (2014). Ongoing Human Action Recognition with Motion Capture. *Pattern Recognition*, 238-247.

Mazzia, V., Angarano, S., Salvetti, F., Angelini, F., & Chiabergea, M. (2022). Action Transformer: A Self-Attention Model for Short-time Pose-Based Human Action Recognition. *Pattern Recognition*, 108487.

Messina, N., Amato, G., Carrara, F., Gennaro, C., & Falchi, F. (2022). Recurrent Vision Transformer for Solving Visual Reasoning Problems. *International Conference on Image Analysis and Processing*, 56-61.

Mohammed G. H. AL Zamil, S. S., Rawashdeh, M., Karime, A., & Hossain, M. S. (2019). Multimedia-Oriented Action Recognition in Smart City-based IoT Using Multilayer Perceptron. *Multimedia Tools and Applications*, 30315-30329.

Neimark, D., Bar, O., Zohar, M., & Asselmann, D. (2021). Video Transformer Network. *International Conference on Computer Vision*, pp. 3163-3172.

Ng, J. Y.-H., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., & Toderici, G. (2015). Beyond Short Snippets: Deep Networks for Video Classification. *Conference on Computer Vision and Pattern Recognition*, 4694-4702.

Pan, C., Yan, W. (2018) A Learning-Based Positive Feedback in Salient Object Detection. International Conference on Image and Vision Computing New Zealand.

Pan, C., Yan, W. (2020) Object Detection Based on Saturation of Visual Perception. Multimedia Tools and Applications, 79 (27-28), 19925-19944.

Pan, C., Liu, J., Yan, W., Zhou, Y. (2021) Salient Object Detection Based on Visual Perceptual Saturation and Two-stream Hybrid Networks. IEEE Transactions on

Image Processing.

Park, S., Park, U., & Kim, D. (2018). Depth Image-based Object Segmentation Scheme for Improving Human Action Recognition. *International Conference on Electronics, Information, and Communication (ICEIC)* , pp.1-3.

Petrushin, V. A. (2005). Mining Rare and Frequent Events in Multi-camera Surveillance Video Using Self-Organizing Maps. *ACM SIGKDD international conference on Knowledge Discovery in Data Mining*, 794-800.

Qiu, Z., Yao, T., & Mei, T. (2017). Learning Spatio-Temporal Representation With Pseudo-3D Residual Networks. *International Conference on Computer Vision*, 5533-5541.

Radhakrishna, A., Yan, W., Kankanhalli, M. (2006) Modeling Intent for Home Video Repurposing. IEEE MultiMedia 13 (1), 46-55.

Rathod, V., Katragadda, R., Saurabh Ghanekar, S. R., Kollipara, P., Rani, I. A., & Vadivel, A. (2020). Smart Surveillance and Real-time Human Action Recognition Using OpenPose. *ICDSMLA 2019*, 504-509.

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. IEEE CVPR, 7263-7271.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv:1804.02767.

Redmon, o., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *Computer Vision and Pattern Recognition* , 779-788.

Ryoo, M. S., Piergiovanni, A. J., Kangaspunta, J., & Angelova, A. (2020). AssembleNet++: Assembling Modality Representations via Attention Connections. *European Conference on Computer Vision*, 654-671.

Saoudi, E. M., & Jai-Andaloussi, S. (2021). A Distributed Content-Based Video Retrieval System for Large Datasets. *Journal of Big Data*, 1-26.

Schuldt, C., Laptev, I., & Caputo, B. (2004). Recognizing Human Actions: A Local SVM Approach. *International Conference on Pattern Recognition*, 32-36.

Sharma, D. K., Chatterjee, M., Kaur, G., & Vavilala, S. (2022). Deep Learning Applications for Disease Diagnosis. In *Deep Learning for Medical Applications with Unique Data* (pp. 31-51). Academic Press.

Shen, D., Chen, X., Nguyen, M., & Yan, W. Q. (2018). Flame Detection Using Deep Learning. *International Conference on Control, Automation and Robotics (ICCAR)*, pp.416-420.

Shen, J., Yan, W., Miller, P., Zhou, H. (2010) Human Localization in a Cluttered Space Using Multiple Cameras. IEEE International Conference on Advanced Video and Signal Based Surveillance.

Shi, Y., Tian, Y., Wang, Y., Zeng, W., & Huang, T. (2017). Learning Long-Term Dependencies for Action Recognition With a Biologically-Inspired Deep Network. *International Conference on Computer Vision (ICCV)*, 716-725.

Shikhar Sharma, R. K. (2015). Action Recognition Using Visual Attention. arXiv:1511.04119 .

Shou, Z., Chan, J., Zareian, A., Miyazawa, K., & Chang, S.-F. (2017). CDC: Convolutional-De-Convolutional Networks for Precise Temporal Action Localization in Untrimmed Videos. *IEEE Conference on Computer Vision and Pattern Recognition*, 5734-5743.

Simonyan, K., & Zisserman, A. (2014). Two-Stream Convolutional Networks for Action Recognition in Videos. *Advances in Neural Information Processing Systems*, 27.

Singh, E., Kuzhagaliyeva, N., & Sarathy, S. M. (2022). Using Deep Learning to Diagnose

Preignition in Turbocharged Spark-Ignited Engines. In *Artificial Intelligence and Data Driven Optimization of Internal Combustion Engines* (pp. 213-237). Elsevier.

Soomro, K., Zamir, A. R., & Shah, M. (2012). UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *Computer Vision and Pattern Recognition*, arXiv:1212.0402.

Sun, L., Jia, K., Chen, K., Yeung, D.-Y., Shi, B. E., & Savarese, S. (2017). Lattice Long Short-Term Memory for Human Action Recognition. *International Conference on Computer Vision*, 2147-2156.

Sun, L., Jia, K., Shen, Y., Savarese, S., Yeung, D. Y., & Shi, B. E. (2018). Coupled Recurrent Network (CRN). arXiv:1812.10071.

Sun, L., Jia, K., Yeung, D.-Y., & Sh, B. E. (2015). Human Action Recognition Using Factorized Spatio-Temporal Convolutional Networks. *International Conference on Computer Vision*, 4597-4605.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2015). Going Deeper With Convolutions. *IEEE CVPR*, 1-9.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning Spatiotemporal Features With 3D Convolutional Networks. *International Conference on Computer Vision*, 4489-4497.

Tripathi, R. K., Jalal, A. S., & Agrawal, S. C. (2017). Suspicious Human Activity Recognition: A Review. *Artificial Intelligence Review*, 238-339.

Ullah, A., Ahmad, J., Muhammad, K., Sajjad, M., & Baik, S. W. (2017). Action Recognition in Video Sequences Using Deep Bi-Directional LSTM With CNN Features. *IEEE Access*, 1155-1166.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . .

Polosukhin, I. (2017). Attention is All you Need. *Advances in Neural Information Processing Systems*, 30.

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2022). YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-art for Real-time Object Detectors. arXiv:2207.02696.

Wang, C.-Y., Yeh, I.-H., & Liao, H.-Y. M. (2021). You Only Learn One Representation: Unified Network for Multiple Tasks. arXiv:2105.04206.

Wang, H., & Schmid, C. (2013). Action Recognition with Improved Trajectories. *International Conference on Computer Vision* , 3551-3558.

Wang, H., Kläser, A., Schmid, C., & Liu, C.-L. (2013). Dense Trajectories and Motion Boundary Descriptors for Action Recognition. *International Journal of Computer Vision*, 60-79.

Wang, J., Yan, W., Kankanhalli, M., Jain, R., Reinders, M. (2003) Adaptive monitoring for video surveillance. International Conference on Information, Communications and Signal Processing.

Wang, J., Kankanhalli, M., Yan, W., Jain, R. (2003) Experiential sampling for video surveillance. ACM SIGMM International Workshop on Video surveillance (pp.77-86).

Wang, J., Bacic, B., & Yan, W. Q. (2018). An Effective Method For Plate Number Recognition. *Multimedia Tools and Applications*, 1679-1692.

Wang, J., Kankanhalli, M. S., Yan, W., & Jain, R. (2003). Experiential Sampling for video surveillance. *ACM SIGMM International Workshop on Video Surveillance*, pp.77-86.

Wang, J., Yan, W.-Q., Kankanhalli, M., Jain, R., & Reinders, M. (2003). Adaptive Monitoring for Video Surveillance. *International Conference on Information,* *Communications and Signal Processing, Pacific Rim Conference on Multimedia*, pp.1193-1143.

Wang, L., Qiao, Y., & Tang, X. (2015). Action Recognition With Trajectory-Pooled Deep-Convolutional Descriptors. *Computer Vision and Pattern Recognition*, pp.4305-4314.

Wang, L., Wang, Z., Xiong, Y., & Qiao, Y. (2015). CUHK&SIAT Submission for THUMOS15 Action Recognition Challenge. *THUMOS Action Recognition challenge*, 1-3.

Wang, L., Xiong, Y., Wang, Z., & Qiao, Y. (2015). Towards Good Practices for Very Deep Two-Stream ConvNets. arXiv:1507.02159.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., & Gool, L. V. (2016). Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. *In European Conference on Computer Vision* , 20-36.

Wang, X., Yan, W. (2019) Human Gait Recognition Based on Self-adaptive Hidden Markov Model. IEEE/ACM Transactions on Biology and Bioinformatics.

Wang, X., Yan, W. (2019) Gait Recognition Using Multichannel Convolutional Neural Networks. Neural Computing and Applications.

Wang, X., Yan, W. (2019) Multi-perspective Gait Recognition Based on Ensemble Learning. Springer Neural Computing and Applications.

Wang, X., Yan, W. (2019) Human Gait Recognition Based on Frame-by-frame Gait Energy Images and Convolutional Long Short-term Memory. International Journal of Neural Systems.

Wang, X., Yan, W. (2020) Non-local Gait Feature Extraction and Human Identification. Springer Multimedia Tools and Applications.

Wang, X., Yan, W. (2020) Cross-view Gait Recognition through Ensemble Learning.

Neural Computing and Applications 32 (11), 7275-7287.

Wang, X., Yan, W. (2022) Human Identification Based on Gait Manifold. Applied Intelligence.

Wang, X., & Yan, W. Q. (2019). Human Gait Recognition Based on Frame-by-Frame Gait Energy Images and Convolutional Long Short-Term Memory. *International Journal of Neural Systems*, 1950027.

Wang, X., Zhang, J., & Yan, W. Q. (2020). Gait Recognition Using Multichannel Convolution Neural Networks. *Neural Computing and Applications*, pp. 14275-14285.

Wentao Hu1, J. Z., Huang, B., Zhan, W., & Yang, X. (2020). Design of Remote Monitoring System for Limb Rehabilitation Training Based on Action Recognition. *Journal of Physics: Conference Series*, 032067.

Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). CBAM: Convolutional Block Attention Module. *European Conference on Computer Vision (ECCV)*, 3-19.

Wu, D., Sharma, N., & Blumenstein, M. (2017). Recent Advances in Video-Based Human Action Recognition Using Deep Learning: A Review. *International Joint Conference on Neural Networks* , 2865-2872.

Wu, F., Wang, Q., Bian, J., Xiong, H., Ding, N., Lu, F., . . . Dou, D. (2022). A Survey on Video Action Recognition in Sports: Datasets, Methods and Applications. arXiv:2206.01038.

Wu, J., Li, Y., Wang, L., Wang, K., Li, R., & Zhou, T. (2019). Skeleton Based Temporal Action Detection with YOLO. *Journal of Physics: Conference Series*, 022087.

Wu, W., Liu, H., Li, L., Long, Y., Wang, X., Wang, Z., . . . Chang, Y. (2021). Application of Local Fully Convolutional Neural Network Combined with YOLOv5 Algorithm in Small Target Detection of Remote Sensing Image. *PloS One*, e0259283.

Wu, X. w., Sahoo, D. y., & C.H.Hoia, S. (2020). Recent Advances in Deep Learning for Object Detection. *Neurocomputing*, 39-64.

Xu, S., Wang, J., Shou, W., Ngo, T., Sadick, A.-M., & Wang, X. (2021). Computer Vision Techniques in Construction: A Critical Review. *Archives of Computational Methods in Engineering*, 3383-3397.

Yan, W., Kankanhalli, M., Wang, J., Reinders, M. (2003) Experiential Sampling for Monitoring. ACM SIGMM Workshop on Experiential Telepresence, 70-72.

Yan, W. Q. (2019). *Introduction to Intelligent Surveillance: Surveillance Sata Capture, Transmission, and Analytics.* Springer.

Yan, W. Q. (2021). *Computational Methods for Deep Learning: Theoretic, Practice and Applications.* Springer.

Yan, W., Kieran, D. F., Rafatirad, S., & Jain, R. (2011). A Comprehensive Study of Visual Event Computing. *Multimedia Tools and Applications*, 443-481.

Yan, W.-Q., & Kankanhalli, M. (2003). Motion Trajectory Based Video Authentication. *International Symposium on Circuits and Systems*.

Yan, X. W. (2020). Non-local Gait Feature Extraction and Human Identification. *Multimedia Tools and Applications*, 6065-6078.

Yang, K., Qiao, P., Li, D., Lv, S., & Dou, Y. (2018). Exploring Temporal Preservation Networks for Precise Temporal Action Localization. *AAAI Conference on Artificial Intelligence*.

Yang, L., Zhang, R.-Y., Li, L., & Xie, X. (2021). SimAM: A Simple, Parameter-Free Attention Module for Convolutional Neural Networks. *International Conference on Machine Learning* , 11863-11874.

Yu, J., Jiang, Y., Wang, Z., Cao, Z., & Huang, T. (2016). UnitBox: An Advanced Object Detection Network. *ACM International Conference on Multimedia* , 516-520.

Yu, Z., & Yan, W. Q. (2020). Human Action Recognition Using Deep Learning Methods. *International Conference on Image and Vision Computing New Zealand (IVCNZ)*, 1-6.

Yuan, J., Liu, Z., & Wu, Y. (2011). Discriminative Video Pattern Search for Efficient Action Detection. *Pattern Analysis and Machine Intelligence*, 1728-1743.

Yucer, S., & Akgul, Y. S. (2018). 3D Human Action Recognition with Siamese-LSTM Based Deep Metric Learning. arXiv:1807.02131.

Yunpeng Chen, Y. K. (2018). A^2-Nets: Double Attention Networks. *Neural Information Processing Systems*, 31.

Zhao, Y., Xiong, Y., & Lin, D. (2018). Trajectory Convolution for Action Recognition. *Advances in Neural Information Processing Systems*, 31.

Zhe, L., Yan, W. Q., & Yang, M. L. (2018). Image Denoising Based on A CNN Model. *International Conference on Control, Automation and Robotics*, pp.389-393.

Zheng, K., & Yan, W. Q. (2018). Video Dynamics Detection Using Deep Neural Networks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 140-150.

Zheng, K., Yan, W. Q., & Nand, P. (2017). Video Dynamics Detection Using Deep Neural Networks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 224-234.

Zheng, Z., An, G., & Ruan, Q. (2017). Multi-Level Recurrent Residual Networks for Action Recognition. arXiv:1711.08238.

Zhou, B., Andonian, A., Oliva, A., & Torralba, A. (2018). Temporal Relational Reasoning in Videos. *European Conference on Computer Vision*, 803-818.

Zhu, W., Hu, J., Sun, G., Cao, X., & Qiao, Y. (2016). A Key Volume Mining Deep Framework for Action Recognition. *Computer Vision and Pattern Recognition*, 1991-1999.

Zhu, Y., Yan, W. (2022) Ski Fall Detection from Digital Images Using Deep Learning. ACM ICCCV.

Zhu, Y., Yan, W. (2022) Image-based Storytelling Using Deep learning. ACM ICCCV.